Systèmes de gestion de fichiers et gestion des disques sous Linux

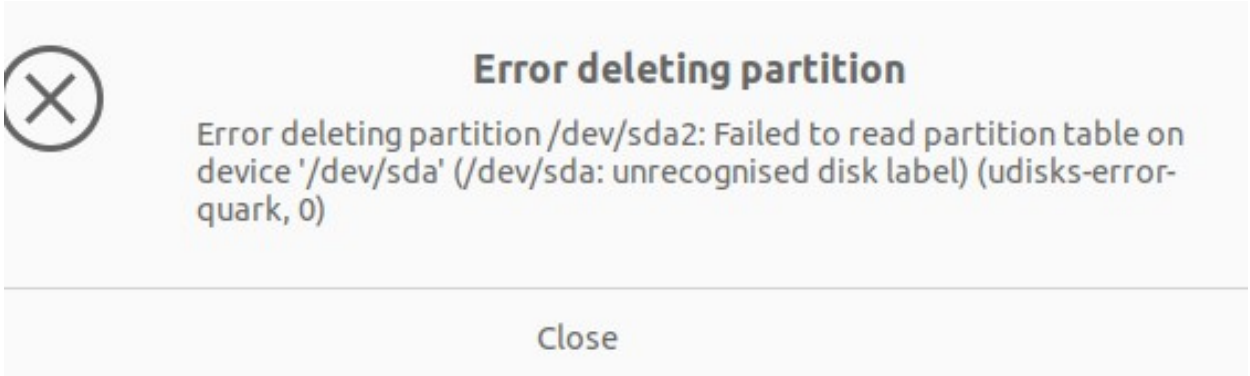
S.berraho@emsi.ma
Berraho.sanae@gmail.com

- Introduction
- Système de fichier
- Représentation d'un fichier
- Organisation générale d'un disque
- Organisation d'un disque sous Linux
- Partitionnement d'un disque sous Linux
- Initialisation du système de fichier
- Montage d'une partition

Introduction

Situation Problématique

RESOLVED
Unable to access location
Error mounting /dev/sda4 at /media/harish/New Volume1: wrong fs type, bad option, bad superblock on /dev/sda4, missing codepage or helper program, or other error
Close
Linux /


Error deleting partition
Error deleting partition /dev/sda2: Failed to read partition table on device '/dev/sda' (/dev/sda: unrecognised disk label) (udisks-error-quark, 0)
Close

Problème :
Votre serveur Linux critique montre :

⇒ **Ralentissements** importants

⇒ **Espace disque** saturé

⇒ **Risque de panne** disque détecté

Mission :
Diagnostiquer et restructurer l'architecture de stockage pour :

⇒ **Améliorer** les performances

⇒ **Sécuriser** les données

⇒ **Garantir** la disponibilité

Objectif : Maîtriser les fondamentaux du stockage Linux pour maintenir un système fiable et performant.

Introduction

Types de disques

HDD

SSD

HDD - Disque Dur Mécanique

- Plateaux magnétiques rotatifs
- Tête de lecture/écriture mobile
- Économique - Grande capacité
- Lent - Sensible aux chocs

SSD - Disque Électronique

- Mémoire flash
- Rapide – Silencieux
- Faible consommation
- Coût supérieur au HDD

Vitesse de Lecture

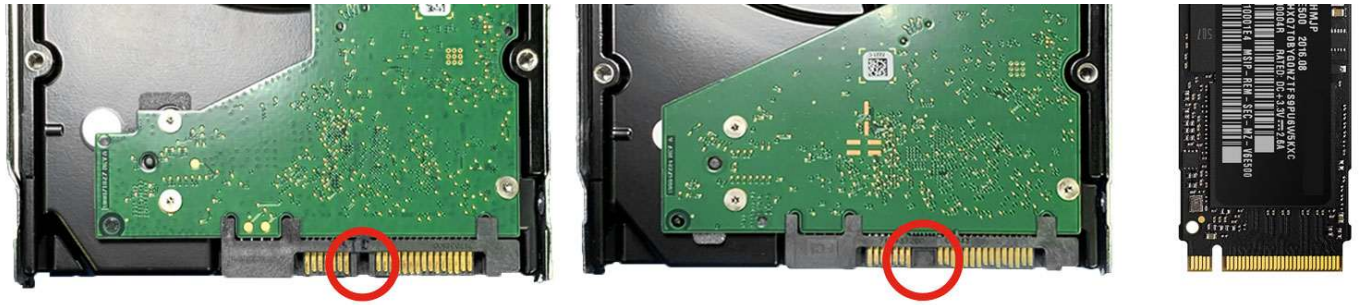
HDD : ~100-200 MB/s
SSD : ~500-3000 MB/s
NVMe : ~3000-7000 MB/s

Temps d'Accès

HDD : 5-10 ms
SSD : 0.1-0.2 ms
NVMe : 0.02-0.05 ms

Introduction

Types d'interfaces



SATA SAS NVMe

SATA (Serial AT Attachment)

- **Usage** : Grand public
- **Prix** : Économique
- **Débit** : Jusqu'à 600 MB/s
- **IOPS** : Jusqu'à 100 000

SAS (Serial Attached SCSI)

- **Usage** : Serveurs entreprise
- **Prix** : Moyen à élevé
- **Débit** : Jusqu'à 1 200 MB/s
- **IOPS** : 200 000 - 400 000

NVMe (Non-Volatile Memory Express)

- **Usage** : Hautes performances
- **Prix** : Moyen à élevé
- **Débit** : 4 000 - 7 000+ MB/s
- **IOPS** : 200 000 - 1 000 000+

Gestion des disques sous Linux

Périphériques Bloc

```
/dev/
├── sda          # Disque SATA/SCSI 1
├── sdb          # Disque SATA/SCSI 2
├── sdc          # Disque SATA/SCSI 3
└── nvme0n1      # Premier disque NVMe
└── nvme0n2      # Deuxième disque NVMe
```

Nomenclature

- /dev/sdX : disques SATA/SCSI
  - sda, sdb, sdc...
- /dev/nvmeXnY : disques NVMe
  - X = contrôleur (0, 1, 2...)
  - Y = namespace (1, 2, 3...)

```
[emsi@localhost ~]$ lsblk | grep disk
nvme0n1     259:0    0   100G  0 disk
sda           8:0    0   500G  0 disk
sdb           8:16   0     1T  0 disk
nvme0n2     259:1    0   500G  0 disk
```

```
[emsi@localhost ~]$ lsblk -d -o NAME,SIZE,TYPE,MODEL
NAME      SIZE   TYPE      MODEL
nvme0n1   100G   disk      Samsung SSD 970 EVO
sda       500G   disk      WDC WD5000LPCX
sdb         1T   disk      Seagate ST1000DM010
nvme0n2   500G   disk      Samsung SSD 980 PRO
```

Gestion des disques sous Linux

Organisation générale d'un disque dur

- Les fichiers sont enregistrés sur des **disques**.
- Les disques sont découpés **en partitions,**
- Chaque disque du système contient **au moins une partition**
- Intérêt d'allouer de l'espace disque sur des **partitions séparées** :

⇒ **Séparation logique des données** du système d'exploitation à partir des données utilisateur

⇒ Capacité à exécuter **plusieurs systèmes d'exploitation sur une machine**

/dev/nvme0n1p1	/boot	1 GB	(boot)
/dev/nvme0n1p2	/	50 GB	(système)
/dev/nvme0n1p3	/var	100 GB	(logs et cache)
/dev/sda1	/home	500 GB	(données utilisateurs)
/dev/sdb1	/backup	1 TB	(sauvegardes)

/dev/sda1	/	(Linux)
/dev/sda2	/windows	(Windows)
/dev/sda3	/data	(Données partagées)

Gestion des disques sous Linux

Partitionnement (1)

- Les disques peuvent être **divisés en partitions**. Chaque partition est accessible comme s'il s'agissait d'un **disque distinct**. Ceci est possible grâce à l'ajout d'une table des partitions.
- **Une table de partitions** permet définir les partitions, avec leur début et fin sur le support de stockage.
- Il existe deux types de tables de partitions:

  ⇒ le type **MBR (Master Boot Record)**
  ⇒ le type **GPT (GUID Partition Table).**

Partitionnement (2)

Tables de partitions: MBR vs GPT

- **MBR** est une ancienne méthode de partitionnement de disque utilisée avec des ordinateurs basés **BIOS**.

- **GPT** est un agencement de partitionnement plus récent faisant partie de l'**UEFI** (Unified Extensible Firmware Interface)

Organisation générale du disque (1)

Partitionnement MBR (Master Boot Record)

- **Le MBR** est situé dans les 1er secteurs du disque
- Sa taille est de 512 octets, Il est constitué de 2 parties :
  ⇒ **La table des partitions (**adresse début et fin de chaque partition**).**
  ⇒ **Le programme d'amorçage** qui charge le noyau du système
- Au démarrage de l'ordinateur, **le BIOS** lit le contenu du MBR du disque placé en première position dans l'ordre d'amorçage.
- Après la lecture du MBR du disque, le BIOS va exécuter **le code d'amorçage** situé dedans.
- L'objectif du **code d'amorçage** est de lancer le chargeur d'amorçage (**bootloader**) du système d'exploitation situé sur la partition du disque marquée comme active.

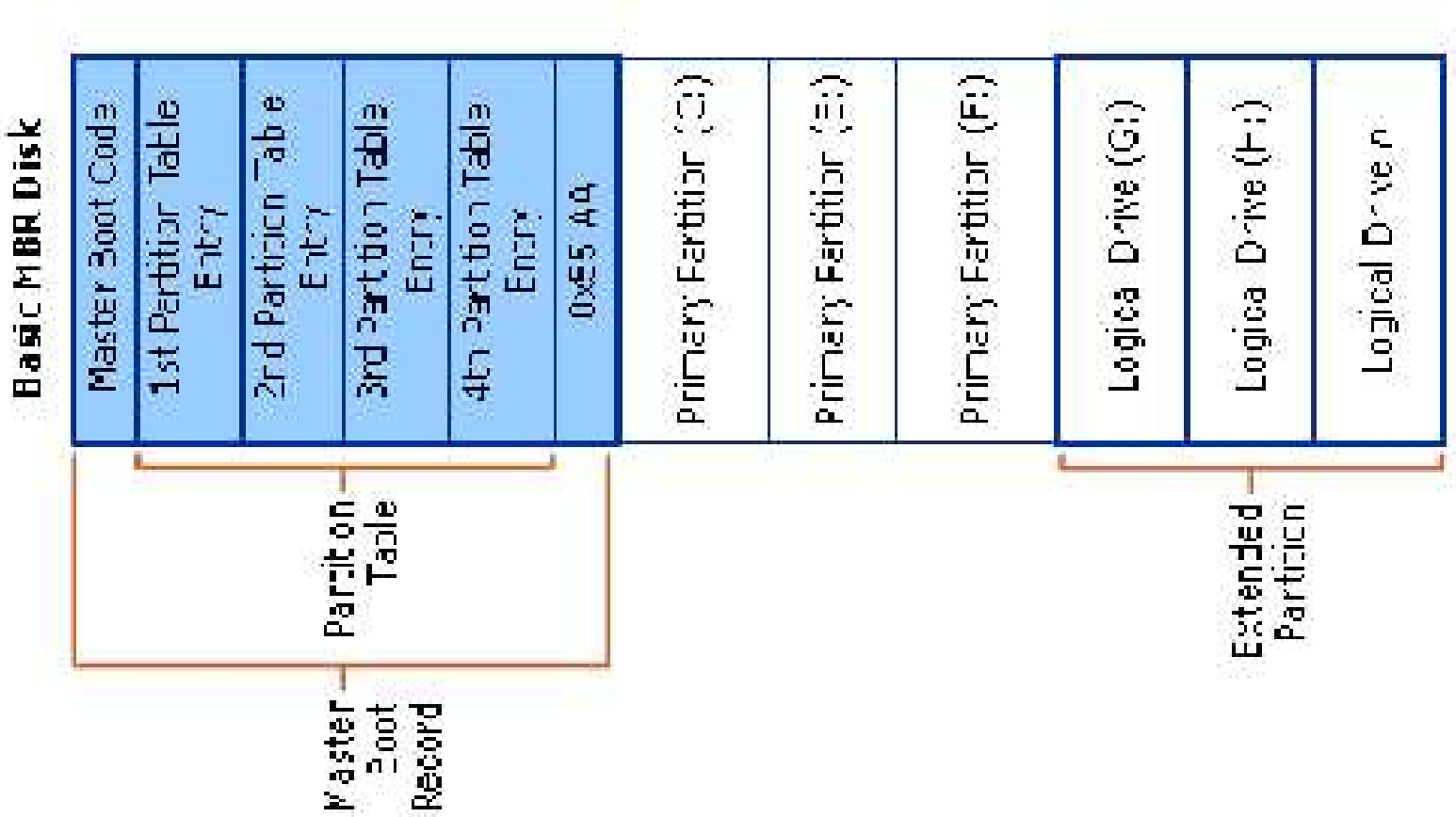
Organisation générale du disque (2)

Partitionnement MBR (Master Boot Record): Types de partitions

- **Partition primaire:**
  ⇒ Une obligatoire et au maximum 4.
- **Partition étendue:**
  ⇒ Une par disque qui remplace une partition primaire (max : 3 primaires + 1 étendue).
  ⇒ Uniquement comme **conteneur** de partitions logiques.
- **Partition logique:**
  ⇒ Contenue dans une partition étendue.

Organisation générale du disque (3)

Structure d'un disque au format MBR



Organisation générale du disque (4)

Partitionnement MBR: Limitations

- Ce système comporte quelques **limitations** qu'il faut connaître si on souhaite encore l'utiliser :
  - ⇒ **La taille des partitions** est limitée à 2,2To.
  - ⇒ On ne peut créer que **quatre partitions principales** (primaires)
  - ⇒ **Incompatible** avec le mode **UEFI**

Organisation générale du disque (5)

Partitionnement GPT (GUI Partition Table)

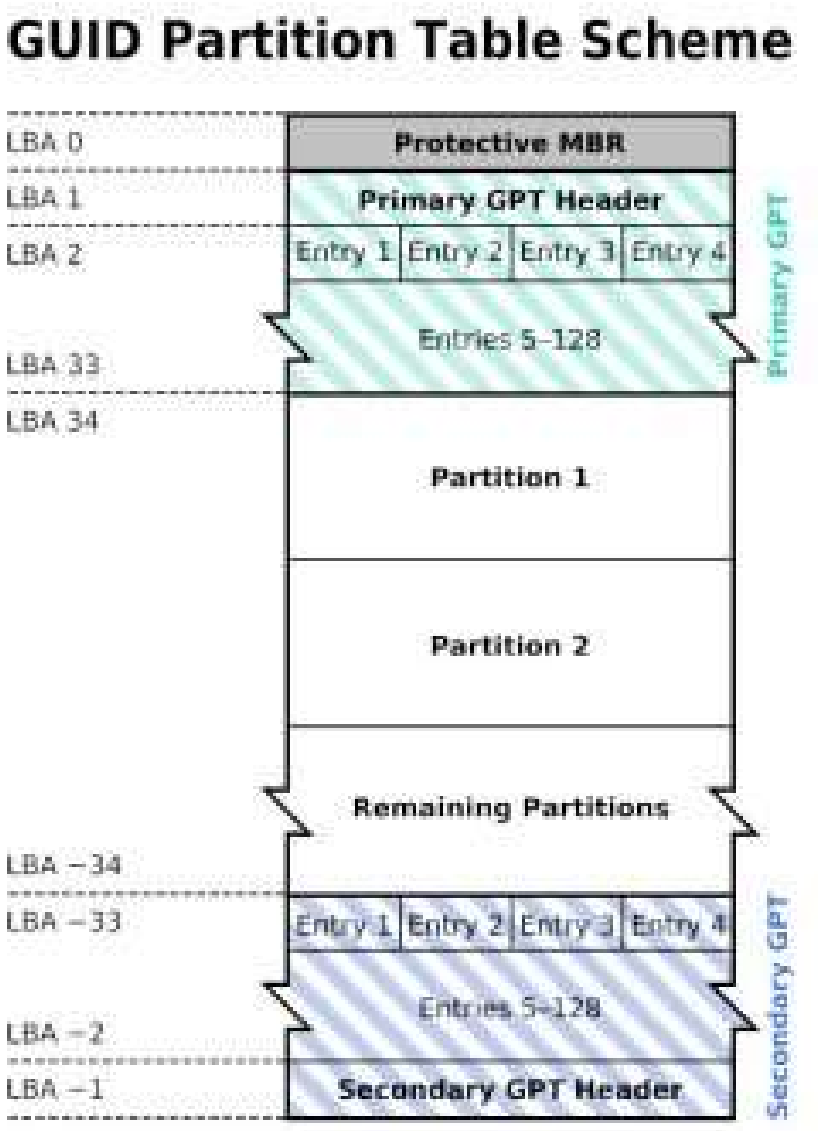
- Chaque partition possède son propre identifiant unique (**GUID**)
- Permet de dépasser les contraintes sur le nombre de partitions
  - ⇒ Il supporte jusqu'à 128 partitions
  - ⇒ La taille des partitions est très grande
  - ⇒ **Compatible** avec le mode **UEFI** et **BIOS**

Si l'on prévoit de créer de **nombreuses partitions physiques** sur le même disque, il convient de créer une table de partition au **format GPT** lors de l'étape du partitionnement.

Organisation générale du disque (6)

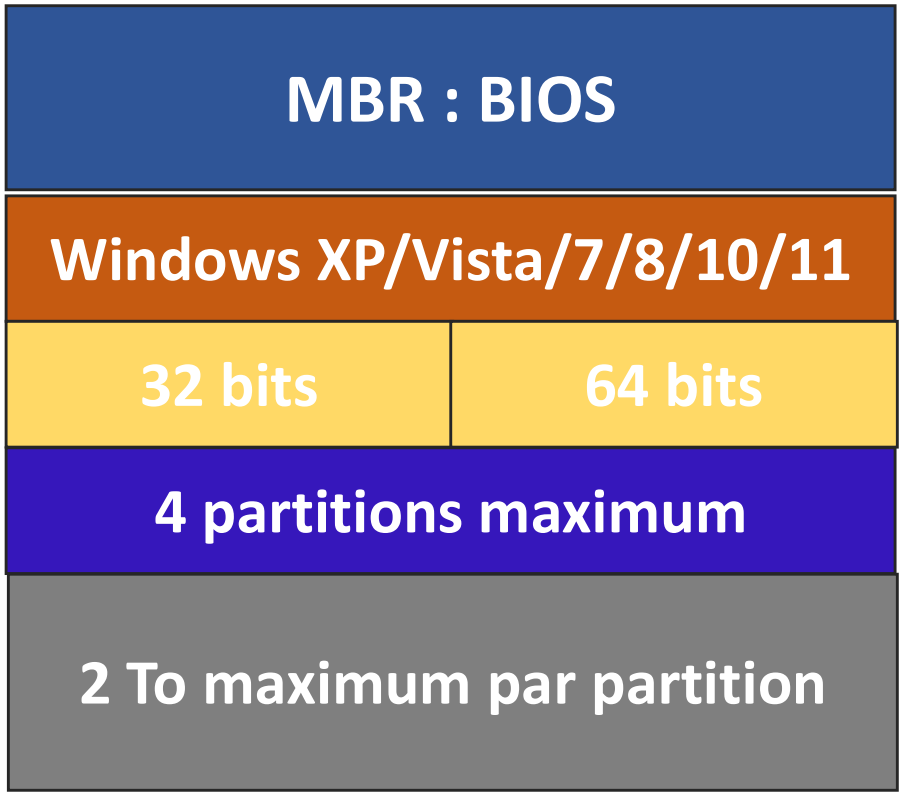
Structure d'un disque au format GPT

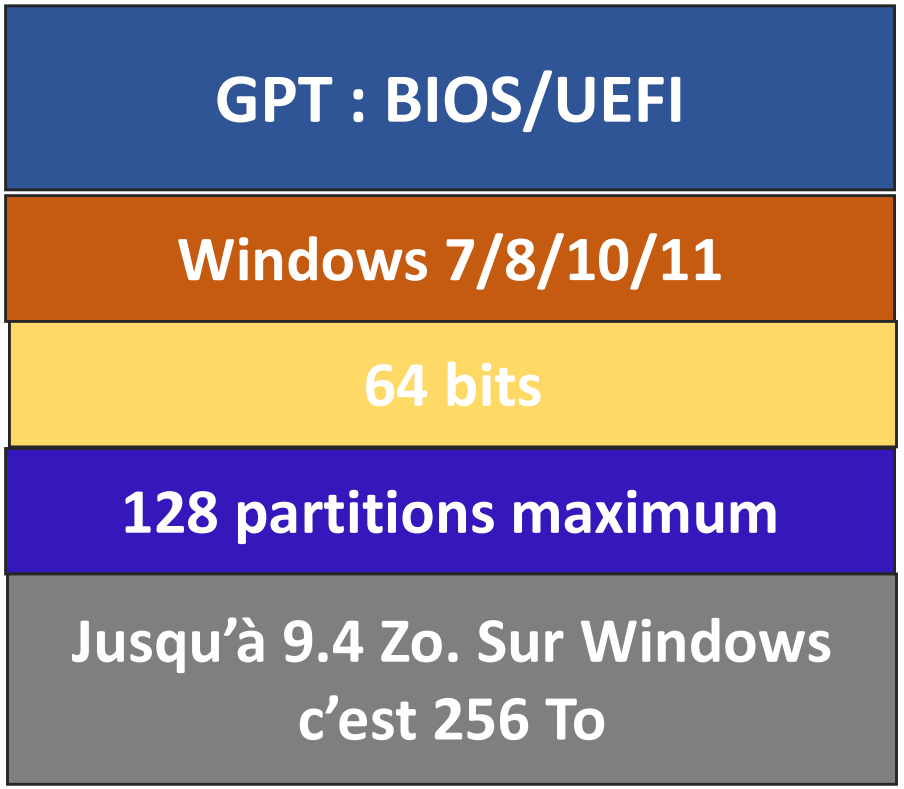
- Pour garantir une compatibilité avec les logiciels gérant uniquement le MBR, le GPT possède **un MBR protecteur (Protective MBR).**
- **Deux GPT sont présents** sur le disque : l'un **primaire**, l'autre **secondaire** (qui est une sauvegarde du premier). Le primaire se situe au début du disque alors que le secondaire se situe à la fin du disque.
- La table des partitions est précédée d'un **entête** qui contient **les informations sur le disque** : blocs utilisables, GUID, nombre et taille des partitions…



Organisation générale du disque (7)

MBR ou GPT ???

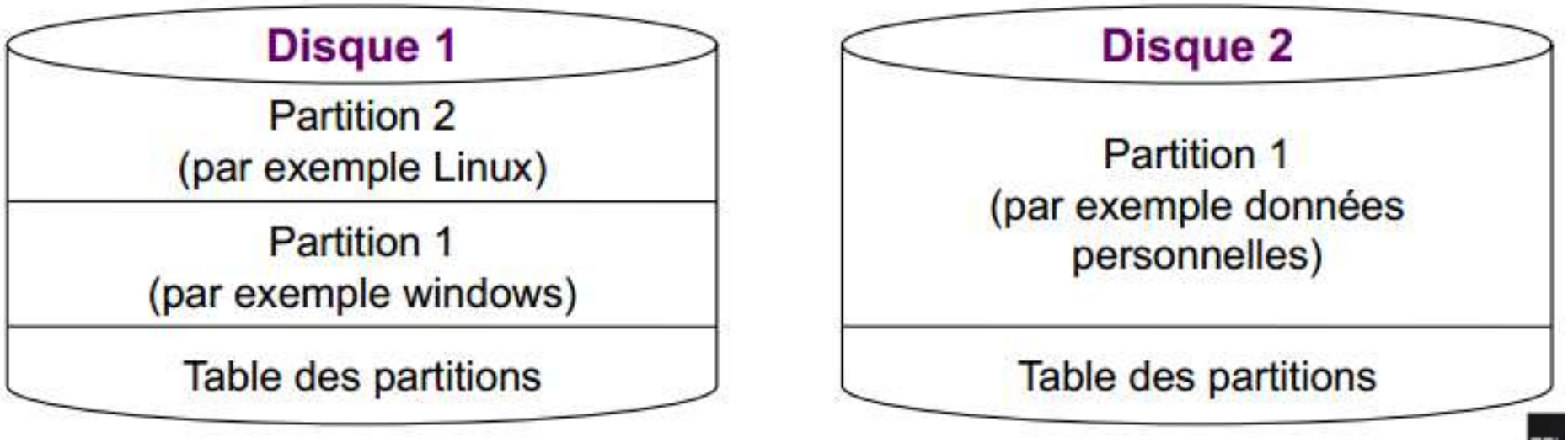




Gestion des disques sous Linux

Organisation des disques sous Linux (1)

- Une machine peut posséder **plusieurs disques**
- Et chaque disque peut être **scindé en plusieurs partitions**
  ⇒ Utile pour installer plusieurs systèmes d'exploitation ou pour augmenter l'indépendance entre les données utilisateurs et le système d'exploitation
- Chaque partition **possède son système de fichiers indépendant**



Gestion des disques sous Linux

Organisation des disques sous Linux (2)

- Le pointeur spécial **/dev** permet l'accès aux disques
- Format des pointeurs sur disque :

- **Types de bus:**
  - ⇒ **hd** : Périphériques IDE
  - ⇒ **sd :** Périphériques SATA, SCSI, SSD, etc.
  - ⇒ **nvme :** Périphériques NVME

- **Exemples:**
  - ⇒ **/dev/hda1** : Partition 1 sur le 1er disque IDE
  - ⇒ **/dev/sdb2** : Partition 2 sur le 2ème disque Sata,SCSI, etc.
  - ⇒ **/dev/nvme0p1**: première partition sur le premier disque NVMe

Gestion des disques sous Linux

Informations sur les disques et les partitions

- Afficher les informations sur les disques et les partitions:

```
# lsblk
```

```
[root@192 bsanae]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                   8:0    0   20G  0 disk
sdb                   8:16   0   20G  0 disk
sr0                  11:0    1  905M  0 rom  /run/media/bsanae/RHEL-8-8-0-BaseOS-x86_64
nvme0n1             259:0    0   20G  0 disk
├─nvme0n1p1         259:1    0    1G  0 part /boot
└─nvme0n1p2         259:2    0   19G  0 part
  ├─rhel_192-root   253:0    0   17G  0 lvm  /
  └─rhel_192-swap   253:1    0    2G  0 lvm  [SWAP]
```

Gestion des disques sous Linux

Informations sur les disques et les partitions

- Au lieu de nommer les partitions ainsi, on préfère maintenant les identifier par **un numéro unique** écrit en hexadécimal

  ⇒ **Universal Unique Identifier:** Identifiant Universel Unique: Il s'agit d'une suite plus ou moins longue de caractères alpha-numériques qui permet d'identifier de façon absolument sûre chaque périphérique de stockage et partition.

  ⇒ Ce n° est inscrit au début de chaque partition

- Pour obtenir l'UUID:

  # blkid

```
[root@192 bsanae]# blkid
/dev/nvme0n1p1: UUID="b8636bc2-1ea6-4ed6-beb0-8acf40e15a1a" BLOCK_SIZE="512" TYPE="xfs" PARTUUID="b242a2b4-01"
/dev/nvme0n1p2: UUID="tN3i0N-Ck2f-5oUV-cpeu-P2AT-vQa4-PxFCiJ" TYPE="LVM2_member" PARTUUID="b242a2b4-02"
/dev/sr0: BLOCK_SIZE="2048" UUID="2023-04-10-21-33-23-00" LABEL="RHEL-8-8-0-BaseOS-x86_64" TYPE="iso9660" PTUUI
/dev/mapper/rhel_192-root: UUID="46c078e5-ad69-43a8-b02f-f0231c9c6440" BLOCK_SIZE="512" TYPE="xfs"
/dev/mapper/rhel_192-swap: UUID="d5075efe-5b93-4774-af03-07e4f6b72de3" TYPE="swap"
/dev/nvme0n1: PTUUID="b242a2b4" PTTYPE="dos"
```

Gestion des disques sous Linux

Informations sur les disques et les partitions

- Afficher la liste des partitions:

```
# ls /dev/
```

```
[root@192 bsanae]# ls /dev/sd*
/dev/sda  /dev/sdb
[root@192 bsanae]# ls /dev/nvme*
/dev/nvme0  /dev/nvme0n1  /dev/nvme0n1p1  /dev/nvme0n1p2
[root@192 bsanae]#
```

- Afficher la liste des partitions et leur taux de remplissage:

```
# df -h
```

```
[root@192 bsanae]# df -h
Filesystem                Size  Used Avail Use% Mounted on
devtmpfs                  1.8G     0  1.8G   0% /dev
tmpfs                     1.8G     0  1.8G   0% /dev/shm
tmpfs                     1.8G  9.8M  1.8G   1% /run
tmpfs                     1.8G     0  1.8G   0% /sys/fs/cgroup
/dev/mapper/rhel_192-root  17G  6.7G   11G  40% /
/dev/nvme0n1p1            1014M  271M  744M  27% /boot
tmpfs                     364M   24K  364M   1% /run/user/1000
/dev/sr0                  905M  905M     0 100% /run/media/bsanae/RHEL-8-8-0-BaseOS-x86_64
```

Gestion des disques sous Linux

Informations sur les disques et les partitions

- Affiche l'espace occupé par une arborescence:

 # du <chemin>

-s: uniquement la somme

-h: un format plus lisible

```
[root@192 bsanae]# du -h /home/bsanae/
0       /home/bsanae/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
0       /home/bsanae/.mozilla/extensions
0       /home/bsanae/.mozilla/plugins
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/3870112724rs
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/3561288849sd
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/1451318868nt
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/2823318777nt
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/1657114595Am
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/2918063365pi
9.3M    /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb
9.3M    /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome
9.3M    /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/temporary
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/default/moz-extension+++f0b0be9b-
ContextId=4294967295/idb/3647222921wleabcEoxlt-eengsairo.files
44K     /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/default/moz-extension+++f0b0be9b-
ContextId=4294967295/idb
48K     /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/default/moz-extension+++f0b0be9b-
```

```
[root@192 bsanae]# du -s /home/bsanae/
103100  /home/bsanae/
[root@192 bsanae]# du -sh /home/bsanae/
101M    /home/bsanae/
[root@192 bsanae]#
```

Gestion des disques sous Linux

Création de partitions (1)

- Pour créer des partitions sous Linux, plusieurs outils existent:

  ⇒ parted,

  ⇒ fdisk,

  ⇒ sfdisk,

  ⇒ gdisk,

  ⇒ sgdisk

Gestion des disques sous Linux

Création de partitions (2)

- Ces outils différent dans le **type de partitionnement**:

  ⇒ **fdisk, sfdisk, parted** sont compatibles avec le partitionnement **MBR** et **GPT**

  ⇒ **gdisk et sgdisk** sont compatibles avec le partitionnement **GPT** seulement, (mais permettent de transformer un partitionnement MBR en GPT et réciproquement)

- Ils différents aussi dans le **mode de fonctionnement**:

  – **fdisk, gdisk** fonctionnent en **mode interactif**
  – **sfdisk, sgdisk** fonctionnent en **mode script**
  – **parted** est utilisable en **mode interactif, script et mixte**

Création de partitions (3)

Création d'une partition: Précautions

- Le partitionnement a pour effet de **créer ou modifier la table de partition et son contenu**, il **ne modifie pas le contenu des partitions.**
- Mais l'écrasement ou la modification malheureuse de la table de partition a pour effet de rendre **inaccessible l'accès aux données**. Il est donc fortement recommandé avant toute opération portant sur la table de partition :
  - ⇒ de sauvegarder les données
  - ⇒ d'identifier correctement les unités de disque ou les partitions à modifier

Création de partitions (4)

Commande fdisk:

- Manipuler la table de partitions d'un disque en mode interactif.

```
# fdisk <disque>
```

- Afficher les partitions d'un disque :

```
# fdisk -l <disque>
```

- Type de partition: Code hexadécimal.
- Exemple :

⇒ **5 :** Partition étendue.

⇒ **7 :** NTFS.

⇒ **b :** FAT32

⇒ **82** : Linux swap

⇒ **83 :** Linux

⇒ **8e :** LVM

- Commandes de **fdisk**:
  - ⇒ **m:** afficher l'aide.
  - ⇒ **p :** lister les partitions.
  - ⇒ **n :** créer une nouvelle partition.
  - ⇒ **d :** supprimer une partition.
  - ⇒ **t :** modifier le type d'une partition.
  - ⇒ **w :** sauvegarder la table de partitions
  - ⇒ **g:** créer une table de partitions de type GPT

Gestion des disques sous Linux

Manipulation

- **Contexte :**
  - ⇒ Un serveur de base de données manque d'espace.
  - ⇒ Vous ajoutez un nouveau disque de 500 Go.
- **Objectif :**
  - ⇒ Créer une partition dédiée aux logs de la base de données.

Étape 1 - Détection du Nouveau Disque

```
# Après insertion du nouveau disque
[admin@greenmarket ~]$ lsblk
NAME      MAJ:MIN RM  SIZE    RO    TYPE      MOUNTPOINT
sda         8:0    0   100G   0     disk
 ├─sda1     8:1    0     1G   0      part      /boot
└─sda2      8:2    0    99G   0      part     /
sdb         8:16   0   500G   0     disk  # ← NOUVEAU DISQUE
```

Gestion des disques sous Linux

Manipulation

Étape 2 - Partitionnement avec fdisk

```
# Lancer fdisk sur le nouveau disque
[admin@greenmarket ~]$ sudo fdisk /dev/sdb
Welcome to fdisk (util-linux 2.37.4).
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-976773167, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-976773167, default 976773167):
Created a new partition 1 of type 'Linux' and of size 500 GiB.

Command (m for help): w
The partition table has been altered.
```

Manipulation

Étape 3 - Vérification de la Partition

```
# Vérifier la nouvelle partition
[admin@greenmarket ~]$ lsblk /dev/sdb
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sdb      8:16   0  500G  0 disk
└─sdb1   8:17   0  500G  0 part

# Voir les détails de la partition
[admin@greenmarket ~]$ sudo fdisk -l /dev/sdb
Disk /dev/sdb: 500 GiB, 536870912000 bytes, 1048576000 sectors
Device     Boot Start       End   Sectors  Size Id Type
/dev/sdb1        2048 976773167 976771120  500G 83 Linux
```

Système de fichiers (1)

- **Un système de fichiers** (file system) définit **l'organisation d'un volume physique ou logique.**

- C'est une structure de données permettant de **stocker les informations et de les organiser dans des fichiers** sur ce que l'on appelle des **mémoires secondaires** (physiquement des **mémoires de masse** comme disque dur, disquette, CD-ROM, clé USB, disques SSD, etc.).

- Un système de fichiers offre à l'utilisateur une vue abstraite sur ses données (**une arborescence de fichiers et de répertoires**) et permet de les **localiser** à partir d'**un chemin d'accès.**

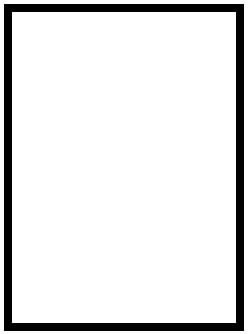
Gestion des disques sous Linux

Système de fichier (2)

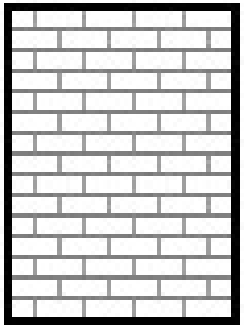
- **Le fichier** est la plus petite entité logique de stockage. **Un nom** lui est associé pour accéder à **son contenu**. Les données du fichier sont stockées dans des suites de **blocs** (la plus petite unité du périphérique de stockage).
- Un système de gestion de fichiers (**SGF**) ou file system (**fs**) est chargé de gérer l'organisation des informations mémorisées sur **les périphériques de bloc** tels que les disques durs, clé USB..
- **Le formatage** (action de formater c'est-à-dire **créer un système de fichiers**) prépare un support de données de stockage en y inscrivant un système de fichiers, de façon à ce qu'il soit **reconnu** par un système d'exploitation.
- Il existe de **nombreux systèmes de fichiers** différents : FAT, NTFS, HFS, ext, UFS, ISO 9660, ReiserFS, APFS, NFS, etc

Gestion des disques sous Linux

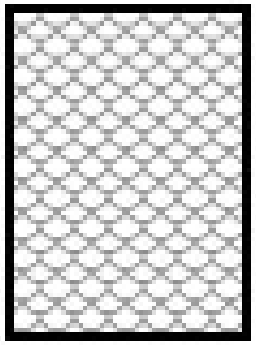
Système de fichier (3)



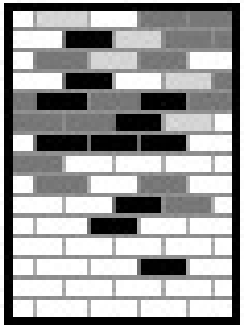
Disque dur non utilisé
(Non formaté)



Disque dur avec un système de fichiers



Disque dur avec un système de fichiers différent



Disque dur sur lequel figurent les données

Extension

- **Le nom d'un fichier** peut posséder un suffixe (**une extension**) séparé par un point qui est fonction du contenu du fichier : .txt pour un fichier texte par exemple.
- **Sous Windows**, cette extension permettra de choisir les applications qui prendront en charge ce format de fichier. Historiquement, l'extension était codée sur 3 caractères dans le système de fichiers FAT utilisé par Microsoft.
- **Sous GNU/Linux**, l'extension fait **simplement partie du nom de fichier**.

Le fichier dans un système de fichiers

- Chaque **fichier** est vu par le système de fichiers de plusieurs façons :
  - ⇒ **un descripteur** de fichier (souvent un entier unique) permettant de l'identifier
  - ⇒ **une entrée dans un répertoire** permettant de le situer et de le nommer
  - ⇒ **des métadonnées sur le fichier** permettant de le définir et de le décrire (attributs)
  - ⇒ **un ou plusieurs blocs** (selon sa taille) permettant d'accéder aux données du fichier (son contenu)

Attributs d'un fichier (1)

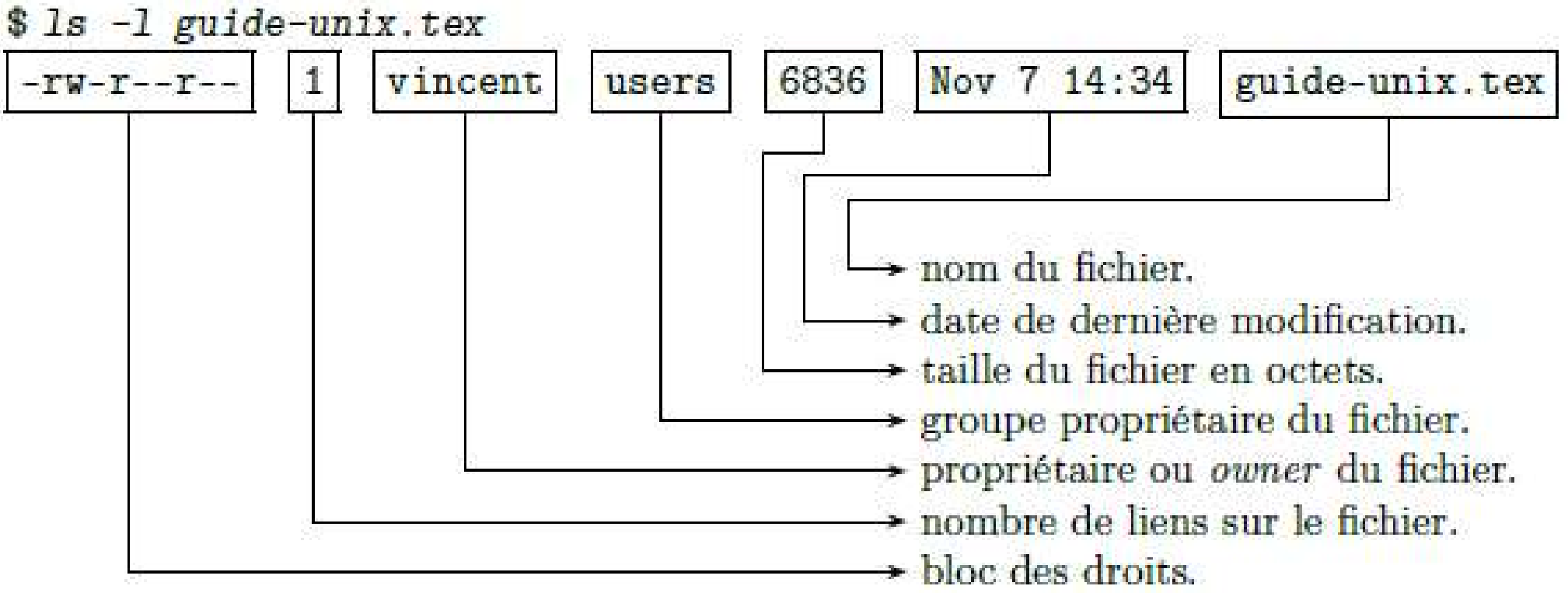
Attributs d'un fichier: Constituent les propriétés du fichier

Exemples d'attributs:

⇒ **Nom:** pour permettre aux personnes d'accéder au fichier
⇒ **Identificateur:** Un nombre permettant au SE d'identifier le fichier
⇒ **Type:** binaire, ou texte; lorsque le SE supporte cela
⇒ **Position:** Indique le disque et l'adresse du fichier sur disque
⇒ **Taille:** en bytes ou en blocs
⇒ **Protection:** Détermine qui peut écrire, lire, exécuter…
⇒ **Date:** pour la dernière modification, ou dernière utilisation
⇒ Autres…

Gestion des disques sous Linux

Attributs d'un fichier (2)

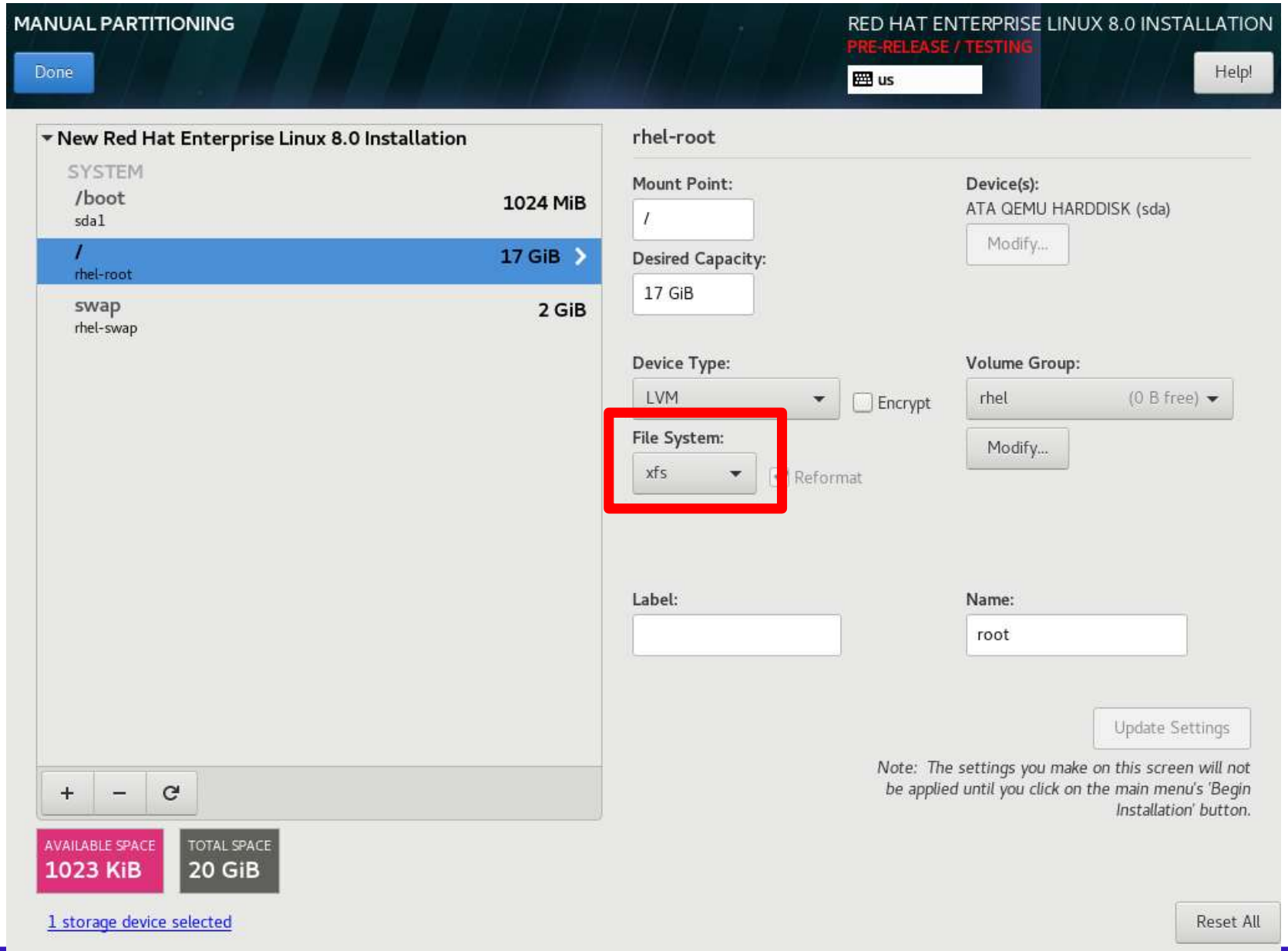


Systèmes de fichiers GNU/Linux (1)

- **L'extended file system** ou **ext**, est le premier S.F. créé en avril **1992** spécifiquement pour GNU/Linux.
- **ext** (1992) : Premier SF dédié Linux
  ext2 (1993) : Stable et robuste - Structure inodes/blocs
  ext3 (2001) : + Journalisation - Récupération après crash
  ext4 (2008) : Volumes > 1 To - Timestamps nanosec - Moins de fragmentation
- **XFS** (1994): Performance fichiers volumineux - Scalabilité élevée
- **Btrfs** (2009): Snapshots et compression - Déduplication données
- **ZFS** (2006): Intégration stockage avancée - Redondance native

Gestion des disques sous Linux

Systèmes de fichiers par défaut dans RHEL 8



Initialisation du système de fichiers

Initialisation du système de fichiers = Formatage

- La création d'un système de fichier (formatage) sur une partition peut se faire avec la commande **mkfs**.

```
# mkfs -t <type> <partition>
```

- Le type détermine la commande à exécuter :
  - ⇒ **ext3** : mkfs.ext3
  - ⇒ **ext4** : mkfs.ext4
  - ⇒ **reiserfs** : mkfs.reiserfs
  - ⇒ **vfat** : mkfs.vfat
  - ⇒ **ntfs** : mkfs.ntfs

- Vérifier le type de système de fichier:

```
# lsblk -f
```

Gestion des disques sous Linux

Système de gestion de fichiers: Définition

- **Système de gestion de fichiers:** ensemble de principes et de règles selon lesquels les fichiers sont **organisés et manipulés.** Chaque système d'exploitation possède son système de fichier privilégié, même s'il peut en utiliser d'autres.

OS	Système de fichier
MSDOS	FAT
Windows 95/98	FAT32 (File Allocation Table 32bits)
Windows NT	NTFS (New Technology File System) standard pour les ordinateurs sous Windows ¿ depuis VISTA.
MAC OS	HFS+ (High Performance File System)
Linux	ext4 (Extended File System)

Représentation de fichiers sur disque (1)

- Les informations d'un fichier sont sauvegardées dans **l'INODE** du fichier (**nœud d'index**) avec d'autres données.

- Deux concepts fondamentaux:
  - ⇒ **Le bloc :** unité de transfert entre le disque et la mémoire (souvent 4096 octets)
  - ⇒ **L'inode (index node)** : descripteur d'un fichier

Un fichier Unix est une suite finie de bytes (octets) Matérialisée par des **blocs disques**, et une **inode** qui contient les propriétés du fichier (**mais pas son nom)**

Représentation des fichiers sur disque (2)

Concept de l'inode

- Le terme **inode** (contraction de **« index »** et **« node »,** en français : nœud d'index) désigne le **descripteur d'un fichier** sous UNIX.
- L'inode d'un fichier est identifié par un numéro (**i-number**) dans le système de fichiers et contient les **métadonnées du fichier**:
  - ⇒ **Type de l'inode** (fichier ordinaire, répertoire, autres)
  - ⇒ **Propriétaire, droits, dates de création/modification/accès, Taille**
  - ⇒ **Liste des blocs du contenu du fichier**
  - ⇒ **…**
- Tout fichier possède son **unique i-node.**
- Les i-nodes sont tous **de même taille**.
- Donc, dans ce cours : **fichier = inode + blocs du fichier**

```
$ ls -i asup # num d'inode du fichier asup
10224747 asup
```

Gestion des disques sous Linux

Représentation des fichiers sur disque (3)

- La commande **stat** permet d'afficher l'intégralité du contenu de l'**inode**

$ stat *nom_fichier*

- Le nom du fichier
- Sa taille
- Son nombre de blocs
- Son inode

- Ses permissions
- L'UID de son propriétaire
- Le GID de son propriétaire
- Ses mtime, atime et ctime
- ….

```
[root@192 bsanae]# stat file1
  File: file1
  Size: 36              Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 38204600    Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Context: unconfined_u:object_r:user_home_t:s0
Access: 2023-12-03 10:07:43.104753012 +0100
Modify: 2023-12-03 10:07:43.101503099 +0100
Change: 2023-12-03 10:07:43.101503099 +0100
 Birth: 2023-12-03 10:06:19.742330157 +0100
```

Gestion des disques sous Linux

Représentation des fichiers sur disque (4)

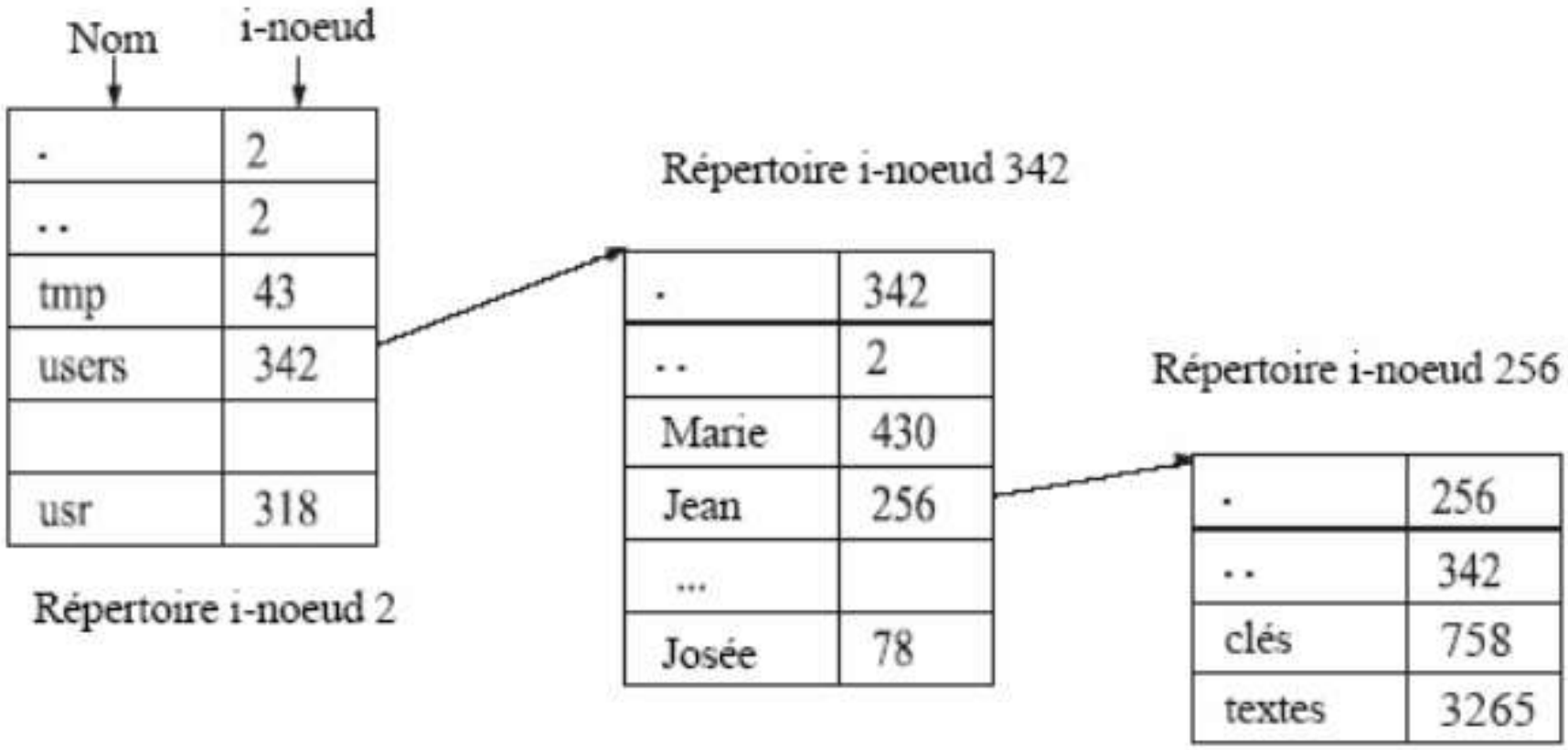
Sous Unix, **un i-node** contient les informations suivantes :

- **Taille du fichier** (en octets) ;
- **Identifiant du disque** qui contient le fichier ;
- **UID :** User IDentifier du propriétaire du fichier ;
- **GID :** Group IDentifier du propriétaire du fichier ;
- **Droits d'accès** (en lecture, écriture, exécution) ;
- **Nombres de liens** (physiques) ;
- **atime (access time) :** Date de dernière ouverture du fichier ;
- **mtime (modification time) :** Date de dernière modification du fichier
- **ctime (change Time) :** Date de dernière modification de l'i-node lui-même ;
- **Adresse :** pointeur sur les blocs du disque sur lesquels le fichier est stocké.

⇒ **Les inodes des fichiers ne contiennent pas les noms des fichiers**

Représentation des fichiers sur disque (5)

Structure d'un répertoire sous Unix



Représentation des fichiers sur disque (6)

Notion de chemin

- **Chemins relatifs :**
  . : répertoire courant ("ici")
  .. : répertoire parent ("au dessus")
  ~ ou **$HOME** : répertoire personnel ("chez moi")

- **Chemin absolu :**
  / : la racine (root)

Représentation des fichiers sur disque (7)

Notion de lien

- **Un lien** est un type spécial de fichiers qui **fait référence à un autre fichier**

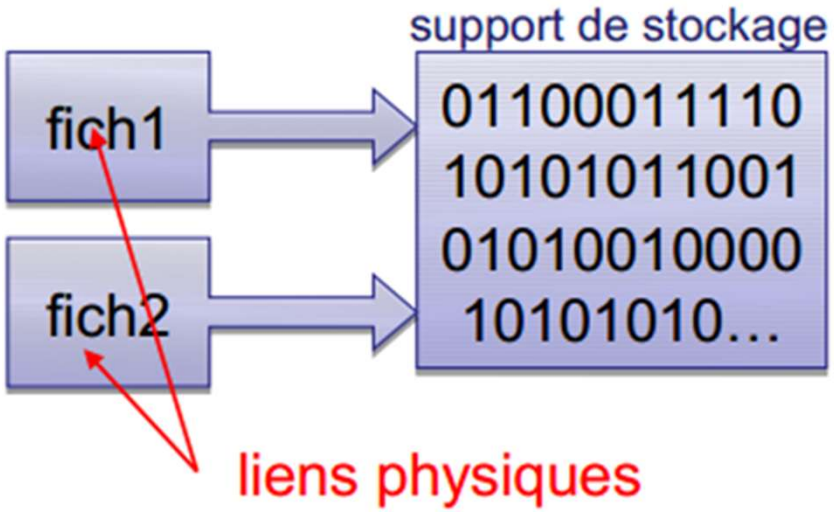
- Le lien permet:
  ⇒ De créer **des raccourcis** vers des fichiers existants
  ⇒ **D'éviter de stocker plusieurs fois le même fichier** dans des répertoires différents

Représentation des fichiers sur disque (8)

Lien physique

- **Un lien physique** est associé à **un emplacement sur le support de stockage.**
- Similaire à la notion de **pointeurs du langage c**
- Deux liens physiques sont considérés comme **deux fichiers indépendants.**
  ⇒ un lien physique permet d'avoir plusieurs noms de fichiers qui partagent exactement le même contenu (**le même i-node**)

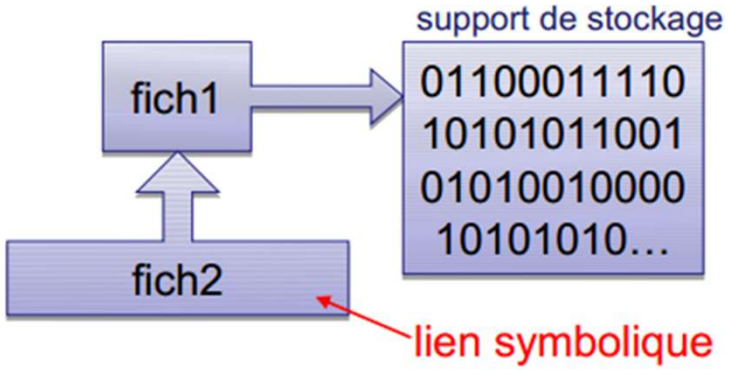
- **Avantage :** On peut changer le contenu ou la localisation du fichier original, le lien physique restera valide.
- **Inconvénient :** un lien physique référence un numéro d'inode ce qui impose de rester dans la même partition (en effet : deux disques logiques différents peuvent avoir chacun une inode 47).
- **syntaxe** : **ln toto ../R2/titi**



Représentation des fichiers sur disque (9)

Lien symbolique

- **Un lien symbolique** est une référence vers un fichier cible
- Utile pour simplifier et réduire l'utilisation du disque quand deux fichiers ont le même contenu.
- Ressemblent plus **aux raccourcis**
- Le principe du lien symbolique est que l'on crée un lien vers un **autre nom de fichier**
- Cette fois le lien pointe **vers le nom du fichier et non pas vers le i-node directement**
- Avec un lien symbolique on peut **traverser les partitions** (plus aucun risque que deux inodes différents aient le même numéro).
- **syntaxe** : **ln –s toto ../R2/titi**
- Par défaut, le lien symbolique doit être **crée dans le répertoire courant**. Si on veut spécifier aussi un chemin relatif pour le lien symbolique, utiliser l'option **r** .



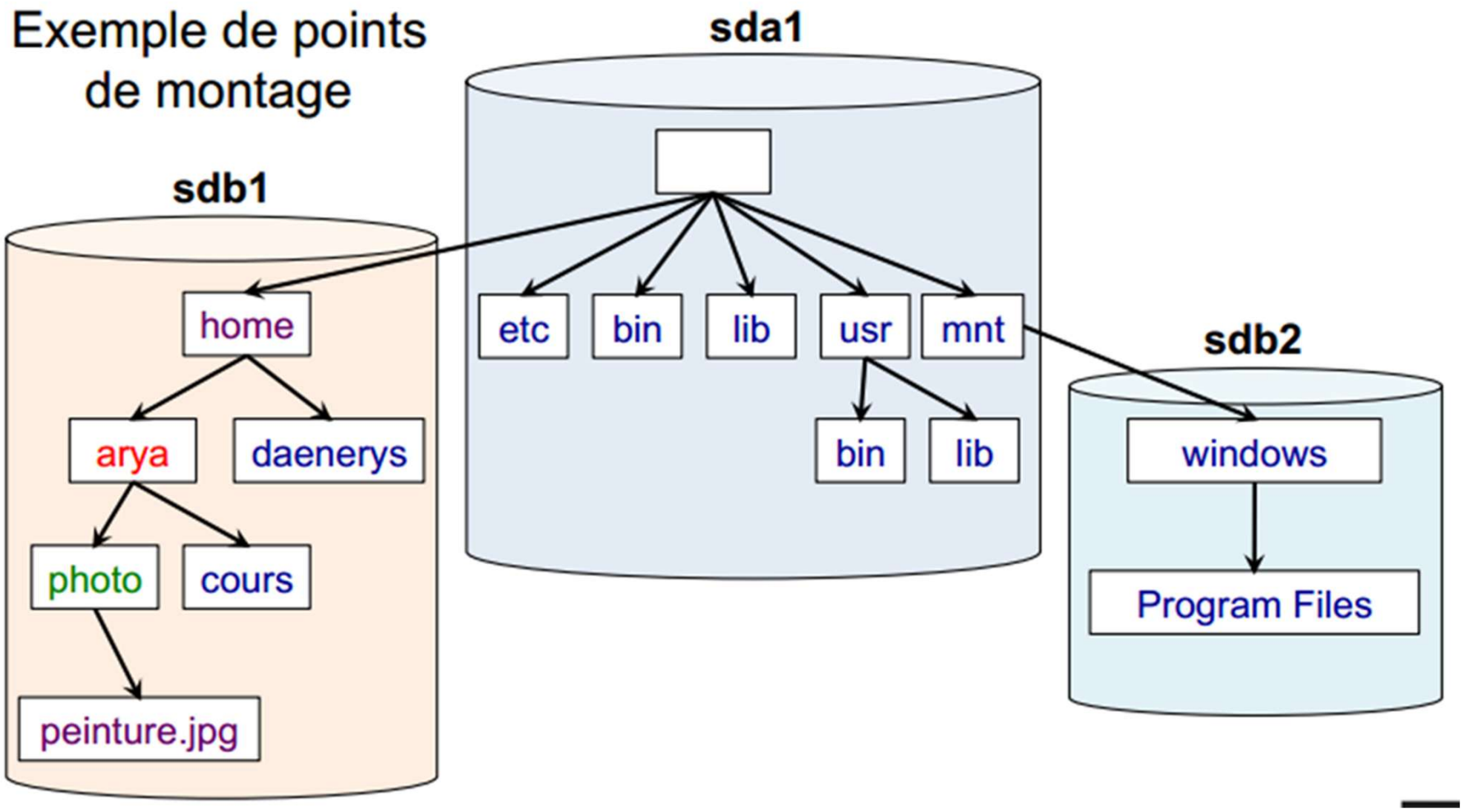
Gestion des disques sous Linux

Montage d'une partition (1)

- L'arborescence Unix peut être construite à partir de **diverses partitions** qui peuvent être situées **sur plusieurs disques**
- Le processus de **montage**, est le moyen de **faire correspondre parties de l'arborescence et partitions physiques de disque**.
- **Un point de montage** est un répertoire à partir duquel sera **accessible** le système de fichiers
- Il suffira ensuite de se déplacer à ce répertoire, appelé **point de montage**, en fait **un répertoire "d'accrochage",** pour accéder à ses fichiers

Gestion des disques sous Linux

Montage d'une partition (2)



Montage d'une partition (3)

Points de montage

- Tant qu'ils ne sont pas effectués, **le système de fichiers est inaccessible**
- Il est possible de créer **un point de montage manuellement**
  ⇒ Pour les clés USB ou les disques durs ou un CDROM par exemple
  ⇒ En utilisant la commande **« mount »**
  ⇒ Et **« umount »** pour supprimer le point de montage

Pour résumer:

⇒ Pour accéder aux fichiers d'une partition, on doit la « **monter** » (**mount**)

⇒ Cela fait **apparaître son contenu dans l'arbre** des fichiers Unix

Montage d'une partition (4)

- **Mount:** Monter un système de fichiers.

```
# mount      partition   point_de_montage
```

- Le point de montage doit être existant. Son contenu (éventuel) deviendra invisible.
- Sans aucun argument **mount** affiche les systèmes de fichiers actuellement montés.
- Options (de la commande) :
  - ⇒ **– t** : Type du SF (détection automatique si pas indiqué).
  - ⇒ **– a** : Monter tous les SF du fichier **/etc/fstab**
  - ⇒ **– L** : Montage par label (étiquette).
  - ⇒ **– U** : Montage par UUID.

Montage d'une partition (4)

- Démonter un système de fichiers.

```
# umount        point_de_montage
```

- On ne peut pas démonter une partition si :
  - ⇒ Une commande s'exécute dans la partition ;
  - ⇒ Un fichier dans cette partition est ouvert ;
  - ⇒ Un répertoire de la partition est le répertoire courant.

- Solution : **lsof**, **fuser**.

Montage d'une partition (5)

Montage automatique: Fichier /etc/fstab

- Contient une configuration statique des différents montages des systèmes de fichiers.
- Permet de faire le montage des systèmes de fichiers **lors du démarrage.**
- Chaque ligne indique :
  - ⇒ Le périphérique à monter ;
  - ⇒ Le point de montage ;
  - ⇒ Le type du système de fichiers ;
  - ⇒ Les options de montage ;
  - ⇒ Fréquences de sauvegarde et de vérification.

Montage d'une partition (6)

Structure du fichier /etc/fstab

Les lignes de **/etc/fstab** contiennent 6 « mots » :

1. Le périphérique associé à la partition, ex : **/dev/sdb1**
2. Le point de montage, ex : **/**
3. Le format de la partition, ex : **ext4**
4. Les options, ex : **defaults, noauto, user**
5. **dump :** activer ou non la sauvegarde du disque avec l'utilitaire dump (1 ou 0)
6. **fsck :** activer ou non la vérification du système de fichier avec l'utilitaire fsck (0, 1, 2 ou 3)

Montage d'une partition (7)

Structure du fichier /etc/fstab

```
#
# /etc/fstab
# Created by anaconda on Sun Oct  1 16:43:12 2023
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel_192-root                   /                       xfs     defaults        0 0
UUID=b8636bc2-1ea6-4ed6-beb0-8acf40e15a1a /boot                   xfs     defaults        0 0
/dev/mapper/rhel_192-swap                   none                    swap    defaults        0 0
```

TP n° 7: Système de gestion de fichiers et gestion des disques sous Linux

Gestion de l’espace Swap sous Linux

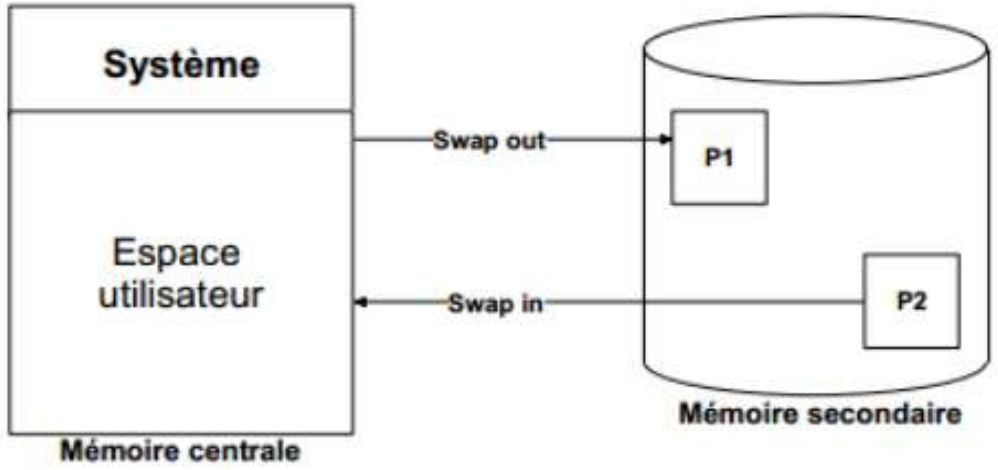
BERRAHO SANAE
berraho.sanae@gmail.com
S.BERRAHO@emsi.ma

Gestion de l'espace swap sous Linux

Zone de Swap

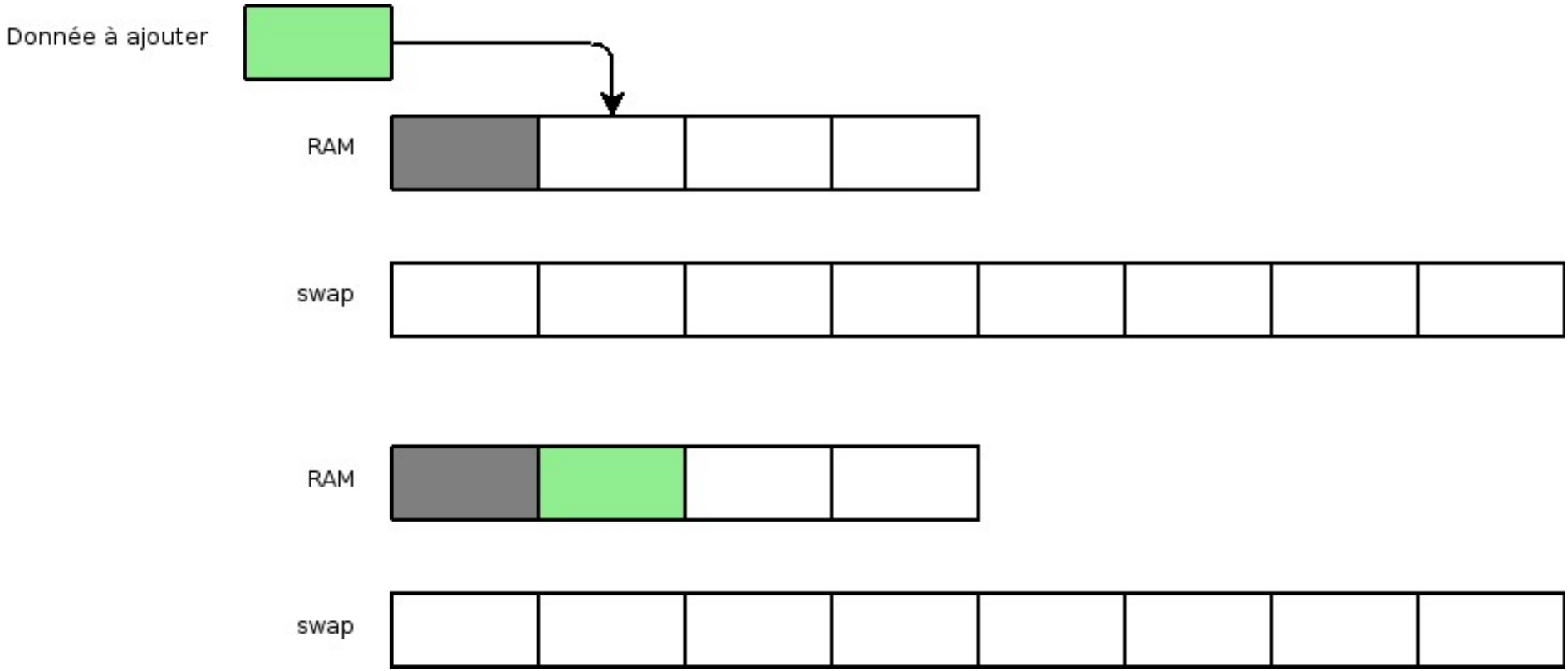
- Avec les systèmes multi-tâches, parfois **la mémoire principale** est insuffisante pour maintenir tous les processus **courant actifs**
- **La zone de swap** est utilisée lorsque **la mémoire physique (RAM) est remplie**. Si le système a besoin de plus de ressources mémoires et que la mémoire physique est remplie, les pages de mémoire (bloc de taille fixe) inactives (**non utilisées depuis un certain temps**) sont déplacées dans la zone de swap
- **Le swapping ou échange** consiste à déplacer temporairement des processus entre la mémoire centrale et la mémoire auxiliaire (disques).

la mémoire swap est la quantité dédiée de disque dur qui est utilisé chaque fois que **la RAM est à court de mémoire.**



Gestion de l’espace swap sous Linux

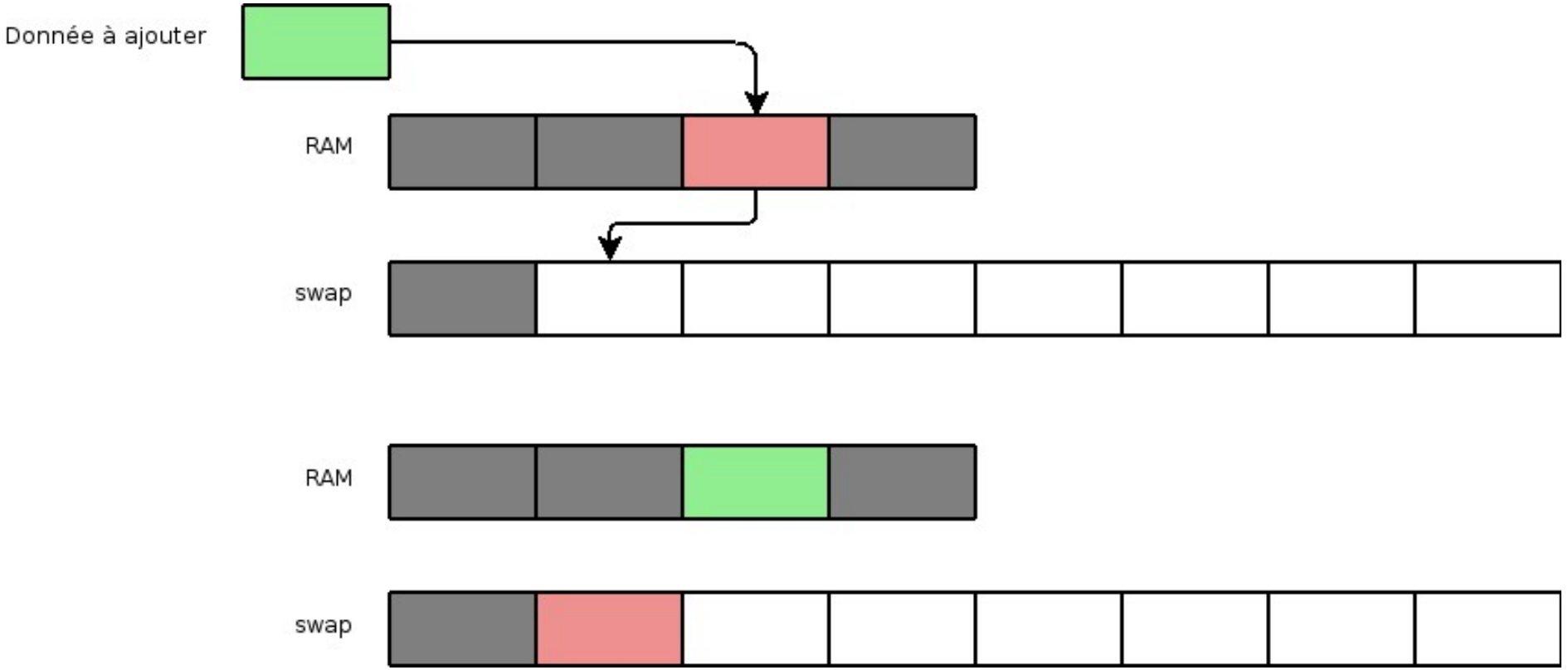
Zone de Swap: Principe de fonctionnement (1)



Tant qu'il y a de la place en mémoire *physique* les données y sont placées

Gestion de l’espace swap sous Linux

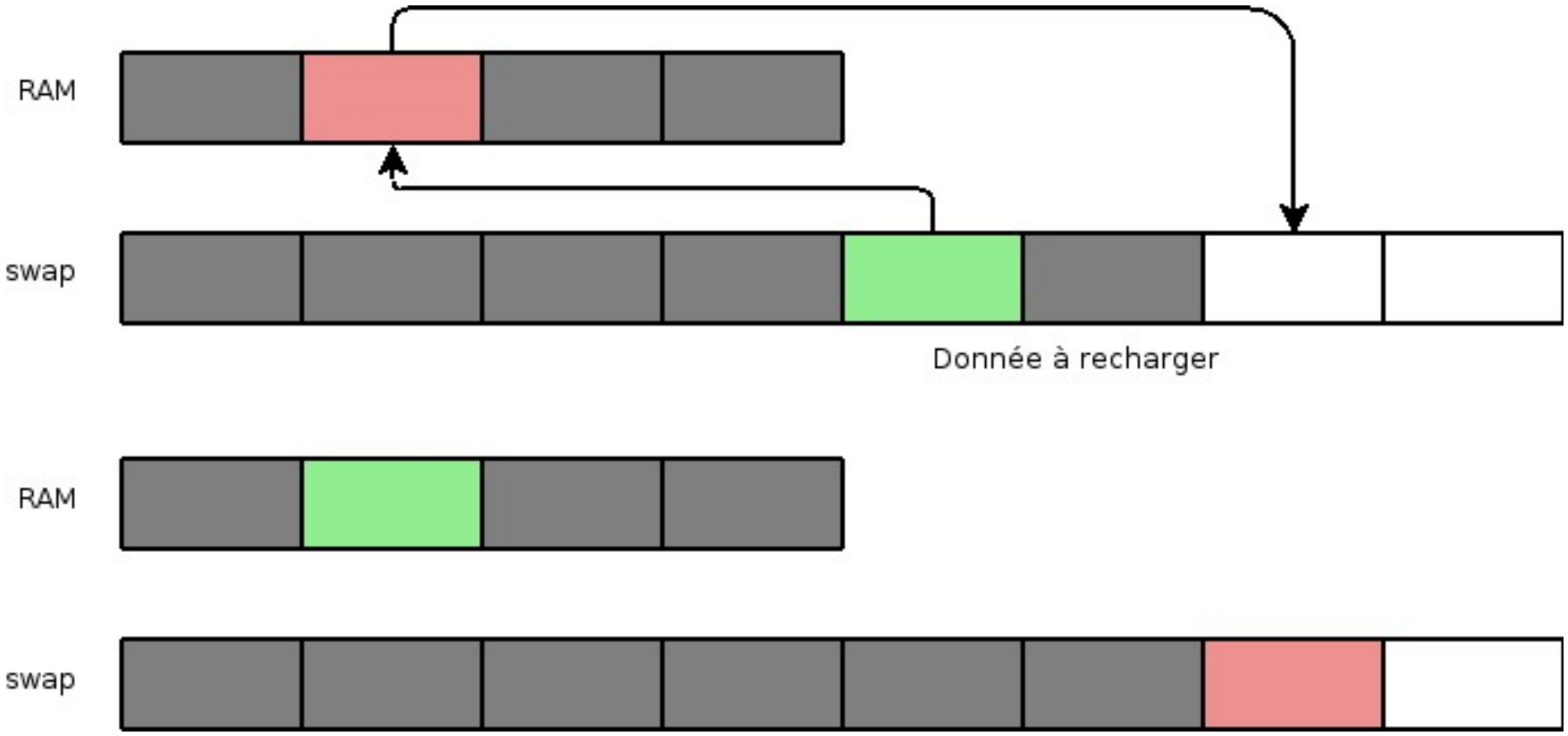
Zone de Swap: Principe de fonctionnement (2)



Le gestionnaire de mémoire virtuelle doit transférer une page dans le fichier d'échange pour libérer de la mémoire physique pour de nouvelles données

Gestion de l’espace swap sous Linux

Zone de Swap: Principe de fonctionnement (3)



Avant de recharger une page en RAM physique, il faut d'abord libérer une page en copiant son contenu vers l'espace d'échange

Gestion de l’espace swap sous Linux

Pourquoi utiliser une zone de Swap?

Même si l'espace swap peut aider les ordinateurs avec une plus petite quantité de RAM, celui-ci ne doit pas être considéré comme un remplacement pour plus de RAM. L'espace swap se trouve sur les disques durs, dont le temps d'accès est plus lent que pour la mémoire physique.

Gestion de l’espace swap sous Linux

Zone de Swap

- La zone de swap peut être stockée dans:

  ⇒ Une partition d’échange

  ⇒ Un fichier swap

Gestion de l'espace swap sous Linux

Configuration du swap sous Linux (1)

Étape 1 - Vérification des informations de swap sur le système

- Vérifier si le système dispose déjà d'un espace d'échange:

```
# swapon --show
```

- Vérifier les échanges actifs :

```
# free -h
```

La commande **free** permet d'afficher l'état de la mémoire RAM du système,

Gestion de l'espace swap sous Linux

Configuration du swap sous Linux (2)

Étape 2 - Vérification de l'espace disponible sur la partition du disque dur

```
# df -h
```

- **Taille optimale à définir pour le swap:**

Quantité de RAM du système	Espace swap recommandé
⩽ 2 Go	2 fois la quantité de RAM
> 2 Go – 8 Go	Égal à la quantité de RAM
> 8 Go – 64 Go	Au moins 4 Go
> 64 Go	Au moins 4 Go

Configuration du swap sous Linux (3)

Étape 3 - Création d'un fichier d'échange

- Générer un fichier de la taille de swap appelé **swapfile** dans le répertoire racine (**/**).

  ⇒ **fallocate :** commande qui crée instantanément un fichier de la taille spécifiée.

```
# fallocate -l 1G /swapfile
```

  ⇒ **dd :**

```
# dd if=/dev/zero of=/swapfile bs=1M count=1024
```

  - **bs** est la taille du bloc
  - **count** est le nombre de blocs
  - **La taille du fichier** = taille de bloc multipliée * count
  - Dans cet exemple de commande dd, le fichier d'échange est de 1 Go (1Mo x 1024)

Configuration du swap sous Linux (3)

⇒ **Vérification :**

```
# ls -lh /swapfile
```

Configuration du swap sous Linux (4)

Étape 4 – Activer fichier d'échange

- Verrouiller les autorisations du fichier afin que seuls les utilisateurs dotés des privilèges * root * puissent en lire le contenu

```
#  chmod 600 /swapfile
```

- Vérifier que les autorisations changent:

```
#  ls –l /swapfile
```

- Marquer le fichier comme espace de swap

```
#  mkswap /swapfile
```

Gestion de l'espace swap sous Linux

Configuration du swap sous Linux (5)

- Activer le fichier d'échange, ce qui permet à notre système de commencer à l'utiliser:

```
# swapon /swapfile
```

- Vérifier que le swap est disponible

```
# swapon --show
```

- Vérifier à nouveau la sortie de l'utilitaire free

```
# free -h
```

Configuration du swap sous Linux (6)

Étape 5 – Rendre le fichier d'échange permanent

- Ajouter les informations du fichier d'échange à la fin de votre fichier /etc/fstab

```
#  gedit /etc/fstab
```

```
/swapfile                none            swap    sw          0       0
```

Gestion de l'espace swap sous Linux

Configuration du swap sous Linux (9)

- Pour mettre hors service une zone de swap, il suffit d'entrer la commande **swapoff** suivie du nom du périphérique:

```
# swapoff /swapfile
```

- Pour supprimer un fichier swap:

1. Supprimer la ligne correspondante du fichier **/etc/fstab**

```
/swapfile                none            swap    sw          0 0
```

2. Supprimer le fichier

```
# rm -v /swapfile
```

Configuration du swap sous Linux (10)

Création d'une partition d'échange

1. Créer une partition (à l'aide de fdisk) destinée à abriter la zone de swap. Préciser le type de la partition (**option t -- > Linux swap**)

2. Marquer la partition comme espace de swap

```
# mkswap /dev/sda6
```

3. Mettre en service la zone swap

```
# swapon /dev/sda6
```

Gestion de l'espace swap sous Linux

Configuration du swap sous Linux (11)

- Ajouter les informations de la partition d'échange à la fin de votre fichier /etc/fstab

```
#  gedit /etc/fstab
```

```
/dev/sda6                none            swap    sw          0 0
```

Gestion des quotas de disques sous Linux

BERRAHO SANAE
berraho.sanae@gmail.com
S.BERRAHO@emsi.ma

Gestion des quotas de disques sous Linux

Problème à résoudre

Gestion des répertoires des utilisateurs

- répertoire des utilisateurs : `/home` (par défaut)
- si un utilisateur occupe toute la place
  - autres utilisateurs lésés
  - si `/home` est sur la même partition que le système : catastrophe ! (ne **jamais** faire ça)

Solution

Utilisation de quotas

- limitation de l'espace disponible par utilisateur
- possibilité de gestion par groupe

Gestion des quotas de disques sous Linux

Définition

- **L'attribution de quotas** dans un système de fichiers est un outil qui permet de **maîtriser l'utilisation de l'espace disque.**

- Les quotas consistent à **fixer une limite d'espace** pour **un utilisateur ou un groupe d'utilisateurs**

- Le but recherché est le plus souvent de contraindre à **une meilleure gestion des répertoires personnels**. Quand un utilisateur dépasse la taille maximale fixée, il est averti et dispose d'un délai pour "faire le ménage" ...

Création de quota: Limites

Pour la création de ces quotas, on définit 2 types de limites :

- **La limite dure (ou hard limit):** définie une limite absolue pour l'utilisation de l'espace. **L'utilisateur ne peut pas dépasser cette limite.** Passée cette limite, l'écriture sur ce système de fichiers lui est interdite
- **La limite douce (ou soft limit):** indique la quantité maximale d'espace qu'un utilisateur peut occuper sur le système de fichiers. Si cette limite est atteinte, **l'utilisateur reçoit des messages d'avertissement quant au dépassement du quota qui lui a été attribué.**
- Si son utilisation est combinée avec les délais (ou **période de grâce**), lorsque l'utilisateur continue à dépasser la soft limite après que le délai de grâce soit écoulé, alors **il se retrouve dans le même cas que dans l'atteinte d'une limite dure.**

Gestion des quotas de disques sous Linux

Création de quota: Ressources disque

- Ces limites sont exprimées en **blocs** et en **inodes.**
- Les quotas exprimés en nombre de blocs représentent **une limite d'espace à ne pas dépasser.**
- Les quotas exprimés en nombre d'inodes représentent **le nombre maximum de fichiers et répertoires que l'utilisateur pourra créer**.
- **Les *délais*** (ou *grace period*) fixent une période de temps avant que la limite douce ne se transforme en limite dure.
- Elle est fixée dans les unités suivantes : **second, minute, hour, day, week.**

Gestion des quotas de disques sous Linux

Mécanisme des quotas

Comment ça marche ?

- Géré par le noyau système (gage d'efficacité)
- Disposer des droits d'administration
- Sur architecture debian : installer le paquet `quota`
- indiquer dans `/etc/fstab` qu'une partition est soumise à quotas.
- Créer des fichiers de configuration à la racine de l'arborescence à placer sous surveillance.

Configuration des quotas de disque

Étape 1 – Installation du paquet quota

- Mise à jour:

```
# dns update
```

- Installation:

```
# dnf install quota
```

- Vérification:

```
# quota --version
```

Étape 2 – Activation des quotas

- Activation des quotas pour les systèmes de fichiers à considérer dans **/etc/fstab**

```
# gedit /etc/fstab
```

- Ajout des paramètres **uquota,gquota** dans la ligne correspondante au système de fichier souhaité:

```
/dev/mapper/rhel-root   /                       xfs     defaults        0 0
UUID=2d66b8c8-31c9-4fc0-94a1-1a500ffd0b7e /boot                   xfs     defaults        0 0
UUID=E04A-BF0B          /boot/efi               vfat    umask=0077,shortname=winnt 0 2
/dev/mapper/rhel-home   /home                   xfs     defaults,uquota,gquota          0 0
/dev/mapper/rhel-var    /var                    xfs     defaults        0 0
/dev/mapper/rhel-swap   none                    swap    defaults        0 0
/swapfile               none                    swap    sw      0       0
```

Étape 3 – Remontage des systèmes de fichiers en question

```
# mount –o remount /home
# systemctl deamon-reload
```

- Vérification de l'utilisation des nouvelles options lors du montage

```
# cat /proc/mounts | grep ' /home '
```

- Résultat attendu:

/dev/mapper/rhel-home /home xfs
rw,seclabel,relatime,attr2,inode64,logbufs=8,logbsize=32k,**usrquota,grpquota**
0 0

Étape 4 – Vérifier l'état des quotas

```
# xfs_quota -x -c "state" /home
```

```
User quota state on /home (/dev/mapper/rhel-home)
  Accounting: ON
  Enforcement: ON
  Inode: #1167 (2 blocks, 2 extents)
Blocks grace time: [7 days]
Blocks max warnings: 0
Inodes grace time: [7 days]
Inodes max warnings: 0
Realtime Blocks grace time: [7 days]
Group quota state on /home (/dev/mapper/rhel-home)
  Accounting: ON
  Enforcement: ON
  Inode: #1174 (2 blocks, 2 extents)
Blocks grace time: [7 days]
Blocks max warnings: 0
Inodes grace time: [7 days]
Inodes max warnings: 0
Realtime Blocks grace time: [7 days]
Project quota state on /home (/dev/mapper/rhel-home)
  Accounting: OFF
  Enforcement: OFF
```

Étape 5 – Attribution des quotas

- **Pour créer un quota sur l'espace disque pour l'utilisateur *ali* :**

```
# xfs_quota -x -c "limit bsoft=2g bhard=3g ali" /home
```

Signification :

- bsoft = 2 Go : limite douce (warning)
- bhard = 3 Go : limite absolue

- **Vérifier:**

```
# xfs_quota -x -c "report -h" /home
```

- **Tester :**

```
# su – ali
# dd if=/dev/zero of=/swapfile bs=1M count=2048
```

Gestion des quotas de disques sous Linux

Configuration des quotas de disque

- Pour créer un quota sur le nombre de fichiers pour l'utilisateur *ali* :

```
# xfs_quota -x -c "limit isoft=5 ihard=8 ali" /home
```

- Vérifier:

```
# xfs_quota -x -c "report -hi" /home
```

- Tester :

```
# su – ali
# touch file1 file1 file3 file4 file5 file6
```

- Pour supprimer un quota:

```
# xfs_quota -x -c "limit isoft=0 ihard=0 ali" /home
```

Gestionnaire de volumes logiques LVM

S.berraho@emsi.ma
Berraho.sanae@gmail.com

Gestionnaire de volumes logiques LVM

Problématique

Le problème :

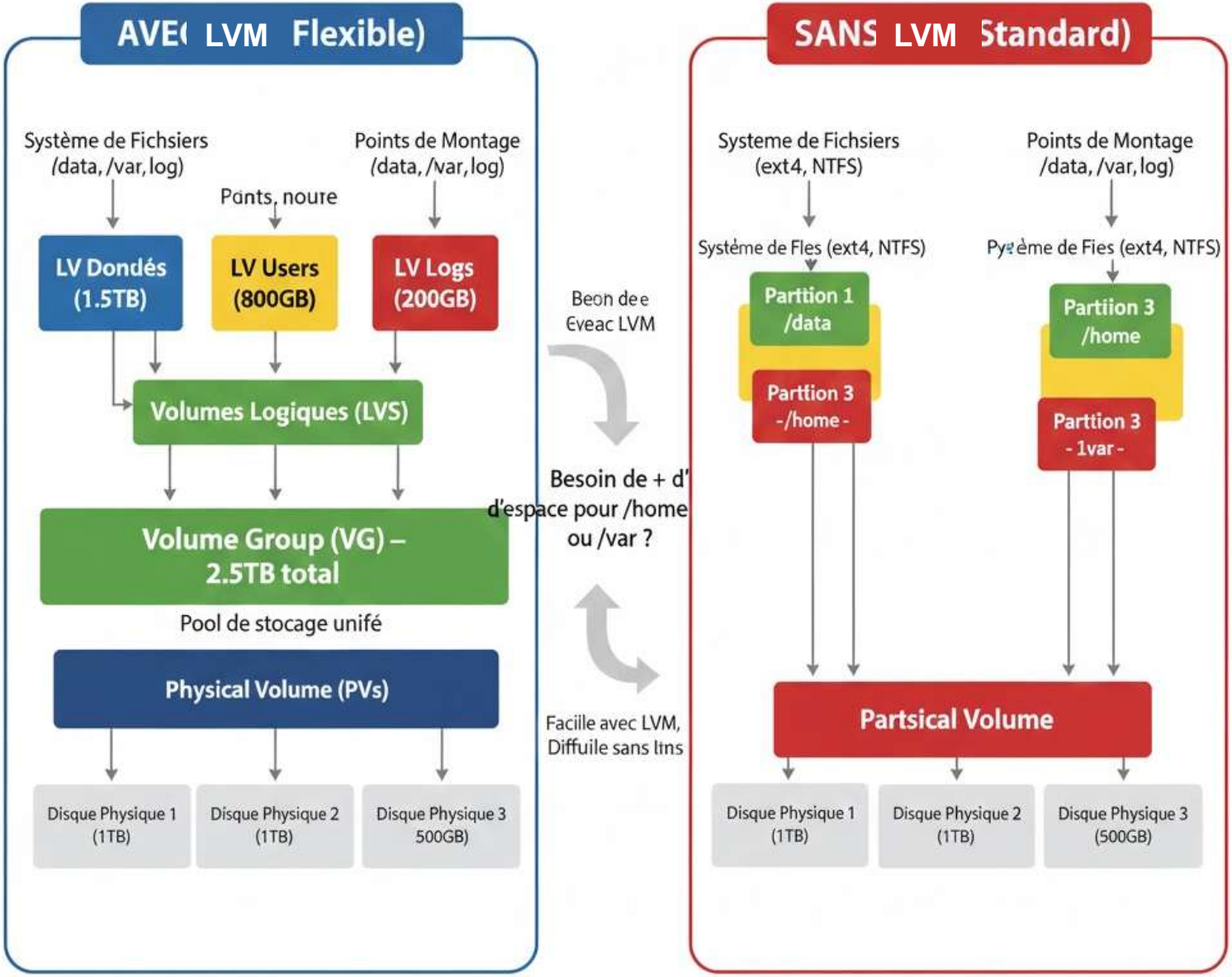
- Partitions classiques = taille fixe
- Impossible de redimensionner facilement
- Gestion rigide de l'espace disque

La solution : LVM

- Abstraction entre disques physiques et systèmes de fichiers
- Gestion flexible de l'espace de stockage
- Redimensionnement à chaud

Gestionnaire de volumes logiques LVM

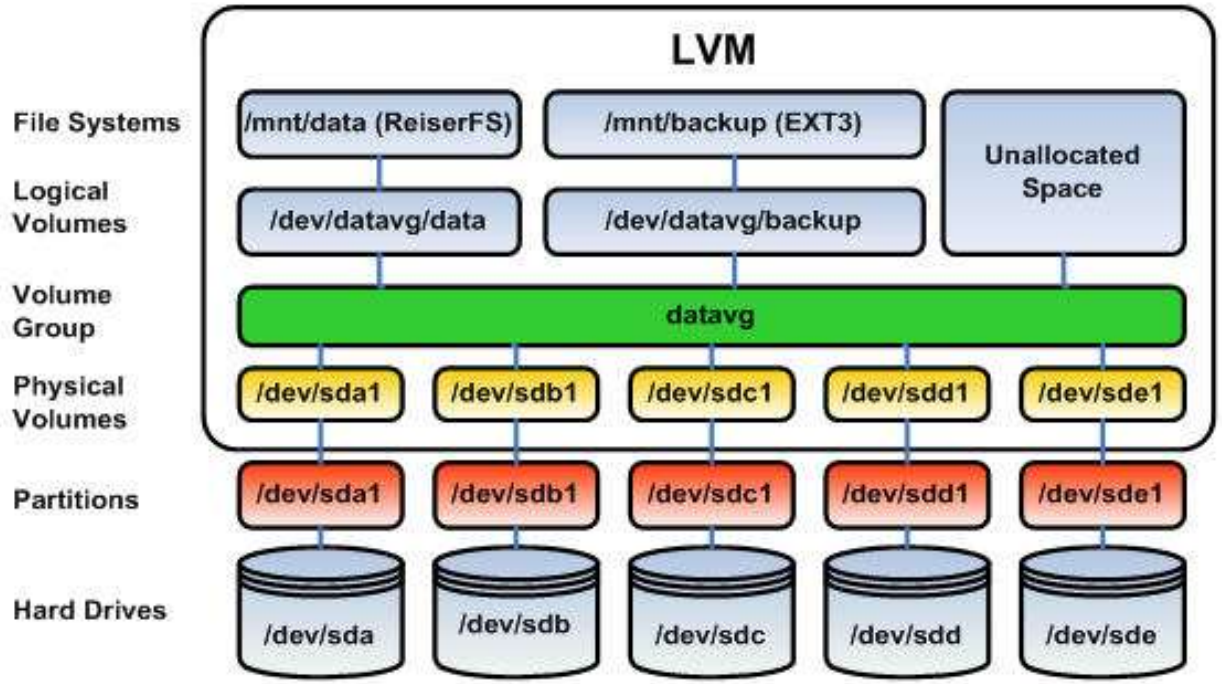
LVM : Logical Volume Manager



Gestionnaire de volumes logiques LVM

LVM : Logical Volume Manager

- **LVM** (Logical Volume Manager, ou gestionnaire de volumes logiques en français) permet la création et la gestion de volumes logiques sous Linux.
- L'utilisation de **volumes logiques** remplace en quelque sorte le partitionnement des disques.
- C'est un système beaucoup **plus souple**, qui permet par exemple de **diminuer** la taille d'un système de fichier pour **pouvoir en agrandir** un autre, sans se préoccuper de leur emplacement sur le disque



LVM : Composants

- **PV : Volumes physiques**

Les disques durs, partitions de disques durs, volumes RAID ou unités logiques provenant d'un SAN forment des « volumes physiques » (physical volumes ou PV).

- **VG : Groupes de Volumes**

On concatène ces volumes physiques (1 ou plusieurs) dans des « groupes de volumes » (volume groups ou VG). Ces VG sont équivalents à des pseudo-disques-durs.

- **LV : Volumes logiques**

Des « volumes logiques » (logical volumes ou LV) sont alors découpés dans les groupes de volumes, puis formatés et montés dans des systèmes de fichiers. Les LV sont équivalents à des pseudo-partitions.

LVM : Logical Volume Manager

1. Pour les systèmes utilisant systemd (comme les dérivées de Red Hat) lancer et activer les services lvm2-lvmetad, lvm2-monitor :

```
systemctl enable  lvm2-lvmetad
systemctl enable lvm2-monitor
systemctl start lvm2-lvmetad
systemctl start lvm2-monitor
```

2. Avant toute chose, il faut créer au moins un PV.
   Un PV peut être un disque entier ou une partition.

```
pvcreate /dev/sdb
```

3. Pour afficher des infos sur un PV, on utilise la commande pvdisplay

```
pvdisplay /dev/sdb
```

Si on ne spécifie pas le nom du PV, tous les PV sont affichés.

LVM : Logical Volume Manager

- Pour supprimer un PV, on utilise la commande **pvremove**

```
pvremove /dev/sdb
```

- Pour créer un VG, on va utiliser la commande **vgcreate**.

```
vgcreate nomduvg /dev/pv1 /dev/pv2
...
```

- Pour afficher des infos sur un VG, on utilise la commande **vgdisplay**

```
vgdisplay datavg
```

- Si le nom du VG n'est pas indiqué, tous les VG sont affichés.
- Pour supprimer un VG, on utilise **vgremove**

```
vgremove datavg
```

LVM : Logical Volume Manager

Ajouter un PV au VG

Si on a un seul PV dans le VG, et qu'on veut agrandir notre VG, on peut ajourer un deuxième PV avec la commande **vgextend** :

```
vgextend datavg /dev/sdc
```

On constate qu'avec **vgdisplay** on a 2 PV maintenant :

```
vgdisplay datavg | grep "Cur PV"
```

Supprimer un PV au VG

Si on a plusieurs PV dans le VG et qu'on veut en ôter un, (donc réduire notre VG) on peut supprimer un PV avec la commande **vgreduce** :

```
vgreduce datavg /dev/sdc
```

Si le nom du VG n'est pas indiqué, tous les VG sont affichés.

Supprimer un VG

Pour supprimer un VG, on utilise **vgremove**

```
vgremove datavg
```

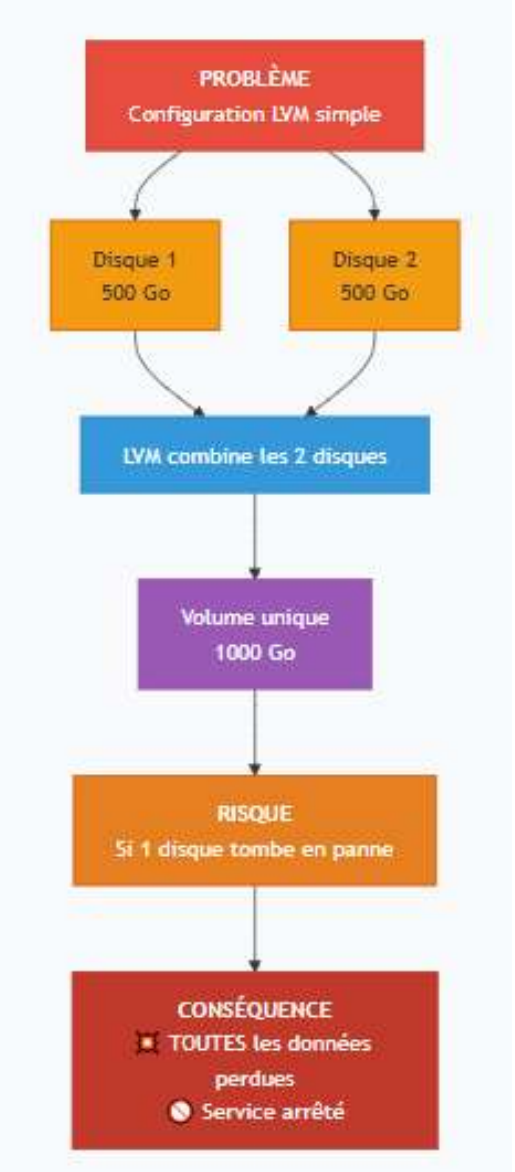
LVM : Logical Volume Manager

Attention Si plusieurs PV dans un même VG

Imaginons 2 disques de 500Go (avec 2 partitions LVM de 500Go) nommés sdb1 et sdc1.

On fait un VG de 1000Go avec ces 2 PV, si 1 des 2 disques tombe en panne, l'intégralité du VG est perdu !

Veiller donc à utiliser **un système de RAID** derrière.



TP n° 8: Système de gestion de fichiers et gestion des disques sous Linux

Systèmes RAIDs

S.berraho@emsi.ma
Berraho.sanae@gmail.com

Systèmes RAIDs

Introduction

- **Les données** constituent **la ressource la plus précieuse** des entreprises d'aujourd'hui.
- **Toute perte de données** se traduit par un manque à gagner.
- Même si vous effectuez **des sauvegardes régulières**, vous avez besoin d'une **méthode de protection intrinsèque** qui garantisse que vos données sont **protégées** et **accessibles sans interruption** en cas de défaillance d'un disque.
- L'ajout d'**un système RAID** à vos configurations de stockage est l'un des moyens les plus économiques pour assurer à la fois **la protection** et **l'accessibilité** de vos données.

Systèmes RAIDs

Introduction

RAID : «**R**edundant **A**rray of **I**ndependant **D**isks »

- Ce système a été inventé en 1987 par trois chercheurs de l'université californienne de Berkeley, dans le but de pouvoir **utiliser des disques de faibles capacités**, donc peu coûteux de telles façons qu'ils soient **vus comme un disque unique**.

- De cette manière, **le RAID** peut être décrit comme étant l'opposé du partitionnement : dans un cas, on crée **plusieurs unités logiques à partir d'un seul disque**, dans l'autre, on crée **une seule unité logique à partir de plusieurs disques physiques**

Systèmes RAIDs

Définition

Un **RAID** est un **regroupement de disques physiques** qui sont employés comme **un disque logique unique**.

⇒ **But:** rehausser la performance et la fiabilité
⇒ **Idée générale:** utilisation simultanée de deux ou plusieurs disques
⇒ **Principe:** Mirroring + Stripping + techniques de correction d'erreurs

- Toutes les implémentations RAID ont en commun **une caractéristique essentielle** : chaque groupe de disques, quel que soit le nombre de disques physiques qui le constituent, peut apparaître à l'utilisateur comme **un disque de plus grande capacité.**

Systèmes RAIDs

Principe

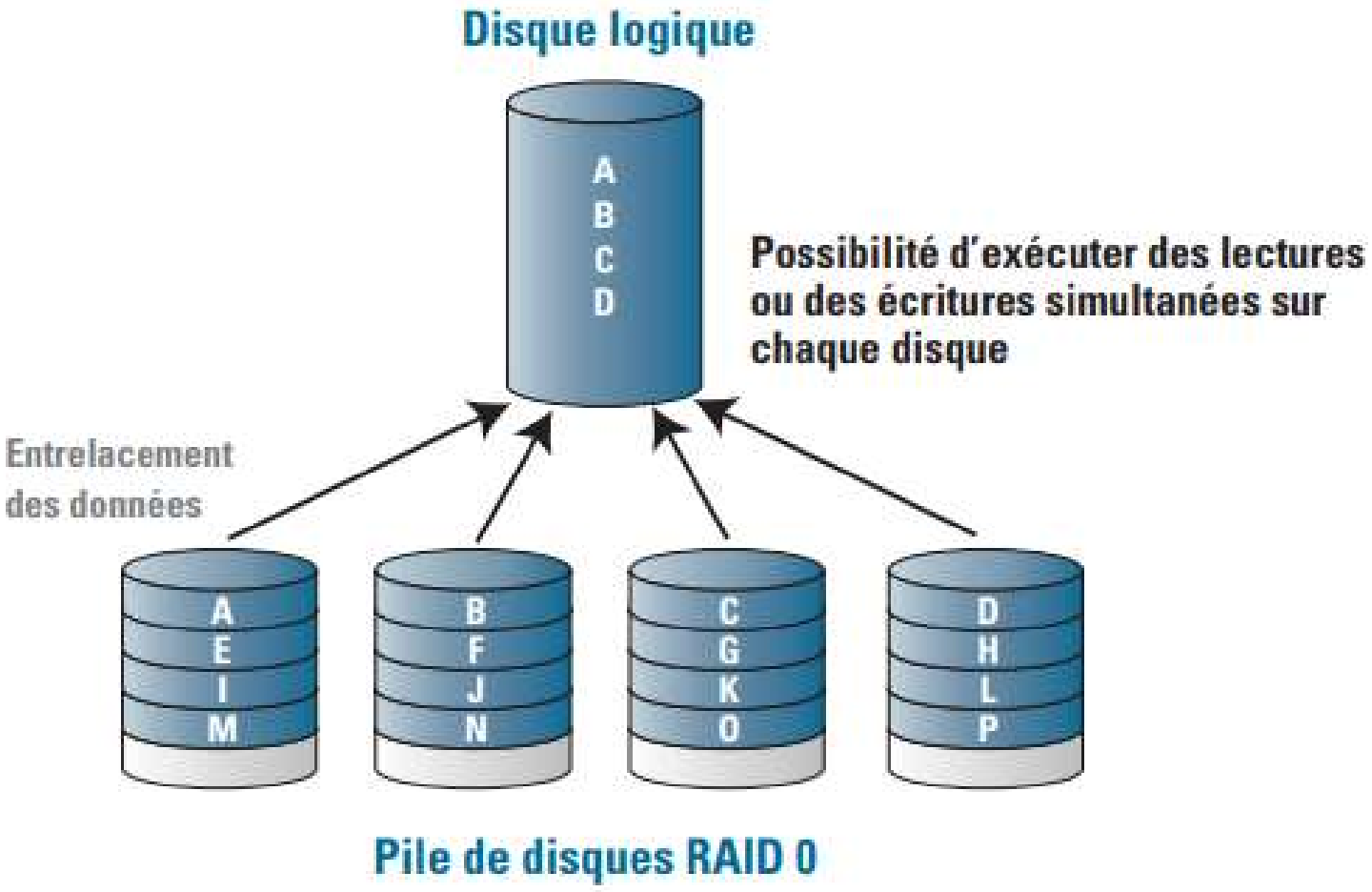
- Le principe de base est la distribution des données sur plusieurs disques d'un même ensemble (**disk array**), les disques étant regroupés en **un seul volume logique**.
- Les données sont découpées en blocs de taille fixe (**chunks**), puis ces blocs sont distribués sur les différents disques du volume logique suivant un algorithme déterminé par **le niveau RAID**.

RAID 0 (1):

Entrelacement par bandes: Stripping

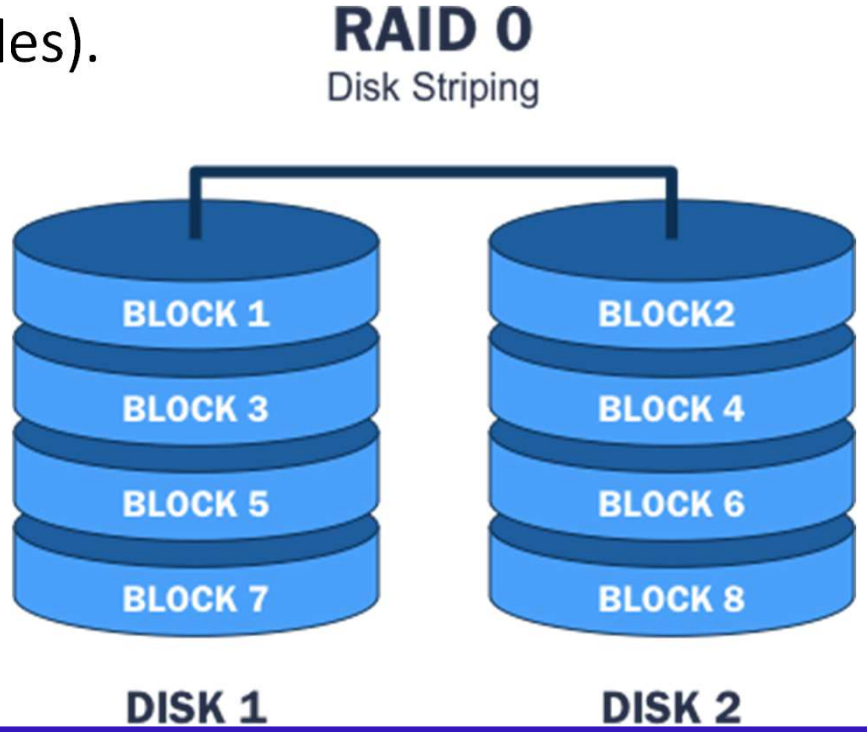
- Combine plusieurs disques en un seul ensemble. Les blocs de données sont répartis sur des bandes de taille **identique réparties uniformément sur les différents disques.**
- Les opérations d'entrées-sorties peuvent donc être **très rapides**, car **effectuées simultanément** par les différents contrôleurs disques.
- En revanche, la **fiabilité de l'ensemble est fortement diminuée**, puisqu'il suffit de perdre un disque pour perdre l'ensemble des données.
- Il n'y a **aucune redondance** des données stockées, et la cohérence des volumes logiques est détruite en cas de défaillance d'un disque.

RAID 0 (2):


Disque logique
A
B
C
D
Possibilité d'exécuter des lectures ou des écritures simultanées sur chaque disque
Entrelacement des données
A
E
I
M
B
F
J
N
C
G
K
O
D
H
L
P
Pile de disques RAID 0

RAID 0 (3):

- L'espace de stockage utile d'un ensemble **RAID 0** est égal à la **capacité utile du plus petit des disques**, multipliée par le nombre de disques qui le composent, puisqu'il n'y a pas de redondance des données et que les bandes de données sont réparties uniformément sur les disques (chaque disque doit avoir le même nombre de bandes).



RAID 0 (4):

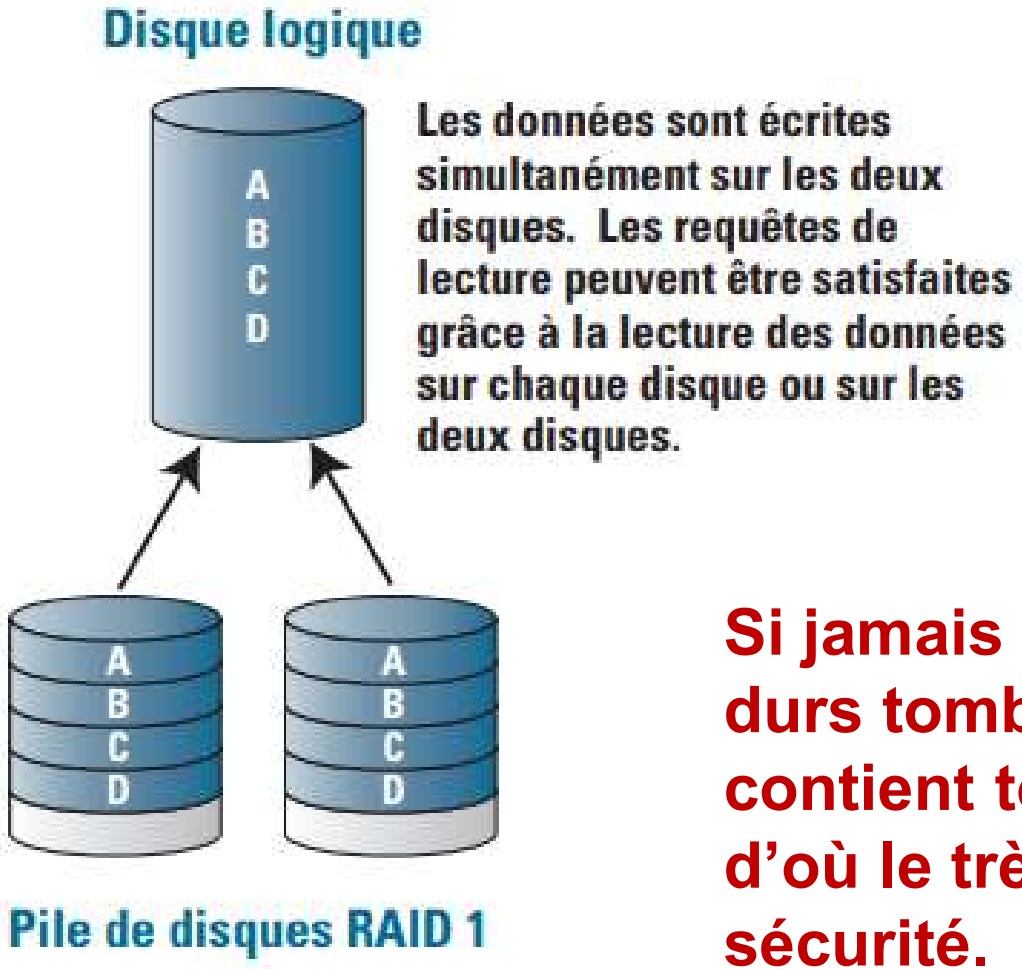
- **Avantages :**
  - ⇒ **Rapidité de lecture et d'écriture** des ensembles de blocs.
  - ⇒ Utilisation **optimale** de l'espace disque, si les disques sont de même taille.
- **Inconvénients :**
  - ⇒ Pas de redondance des données, donc **pas de tolérance de panne.**
  - ⇒ La perte d'un **disque compromet l'ensemble des données stockées**, la fiabilité de l'ensemble est égale à la fiabilité du moins fiable des disques utilisés.

RAID 1 (1):

Disk Mirroring

- **Le RAID 1** (disques miroirs, mirroring) combine plusieurs disques en un seul ensemble.
- Chaque bloc de données utile est **écrit sur chacun des disques**. Cette redondance assure **une excellente fiabilité** à l'ensemble, d'autant plus grande qu'il y a davantage de disques.
- Tant qu'**un seul** disque reste opérationnel, **les données sont intactes**,
- Les opérations de lecture peuvent être plus rapides, car elles peuvent être effectuées simultanément par les différents contrôleurs disques.
- La performance d'écriture se voit dégradée.
- L'espace de stockage utile d'un ensemble **RAID 1** est égal à **la capacité utile du plus petit des disques**

RAID 1 (2):



Si jamais un des deux disques durs tombe en panne, l'autre contient toutes les informations, d'où le très bon niveau de sécurité.

RAID 1 (3):

- **Avantages :**
  ⇒ Excellente **tolérance de panne**, proportionnelle au nombre de disques combinés.
  ⇒ **Bonnes performances** en lecture.

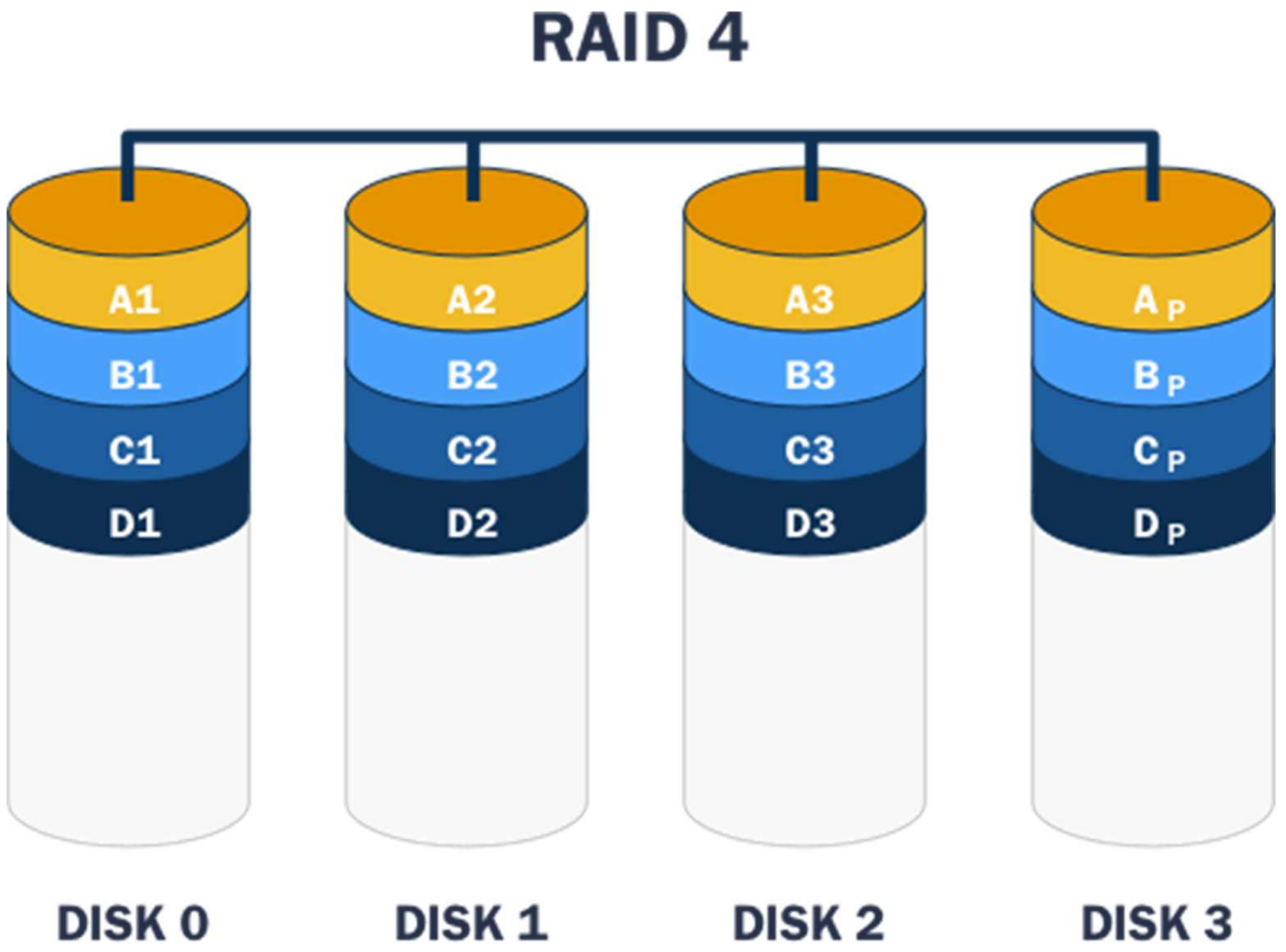
- **Inconvénients :**
  ⇒ **L'espace disque nécessaire** est au moins deux fois la taille de l'espace disque utile.
  ⇒ Les performances en écriture peuvent être impactées, même si en général les écritures se font simultanément sur les différents disques.

Le RAID 4 (1):

Stripping avec contrôle de parité sur disque dédié

- RAID 4 : utilisation du **concept de parité**
- **Pour chaque groupe de *x* disques entrelacés**, ajout d'**un disque supplémentaire** : bit de parité formé à partir des bits des *x* disques
- **Crash d'un disque :** restauration du contenu bit par bit à partir de *x* - 1 autres disques et du disque de parité
- **Écriture de données :** actualisation du disque de parité ➔ **Rapidité inférieure** à RAID 0 et RAID 1

Le RAID 4 (2):

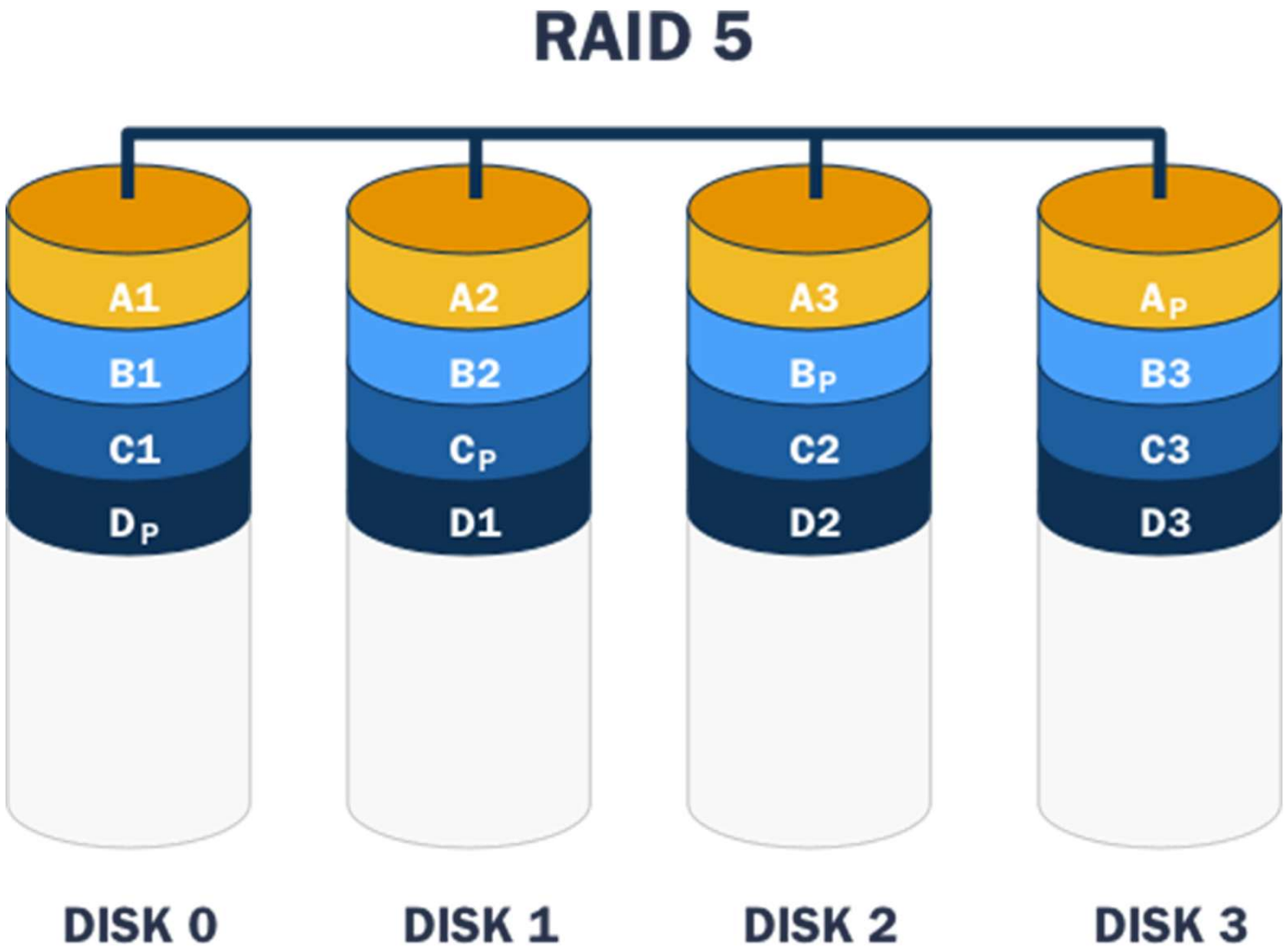

RAID 4
A1
B1
C1
D1
A2
B2
C2
D2
A3
B3
C3
D3
A P
B P
C P
D P
DISK 0
DISK 1
DISK 2
DISK 3

Le RAID 5 (1):

Stripping avec contrôle de parité répartie

- L'architecture RAID de niveau 5 est la même que le RAID de niveau 4 sauf que cette fois les bandes de contrôle (bits de parités) **sont réparties sur tous les disques**
- Le RAID 5 combine **au moins trois disques** en un seul ensemble.
- Pour chaque ensemble de bandes, **une bande de parité est calculée et écrite sur le disque restant.**
- L'emplacement de la bande de parité est **réparti à tour de rôle sur les disques.**

Le RAID 5 (2):


RAID 5
A1
B1
C1
D_P
A2
B2
C_P
D1
A3
B_P
C2
D2
A_P
B3
C3
D3
DISK 0
DISK 1
DISK 2
DISK 3

Le RAID 5 (3):

- En cas de **perte d'une bande de données**, la **bande de parité permet de la reconstituer**, assurant ainsi **la tolérance de panne.**
- Les **opérations de lecture peuvent être très rapides,** car effectuées **simultanément par les différents contrôleurs disques.**
- Les écritures **peuvent être ralenties**, à cause du **calcul et de l'écriture de la bande de parité.**
- **L'espace de stockage utile** d'un ensemble RAID 5 = la capacité du plus petit des disques, multipliée par le nombre de disques qui le composent, moins 1 à cause des **bandes de parité**

Combinaisons de RAID

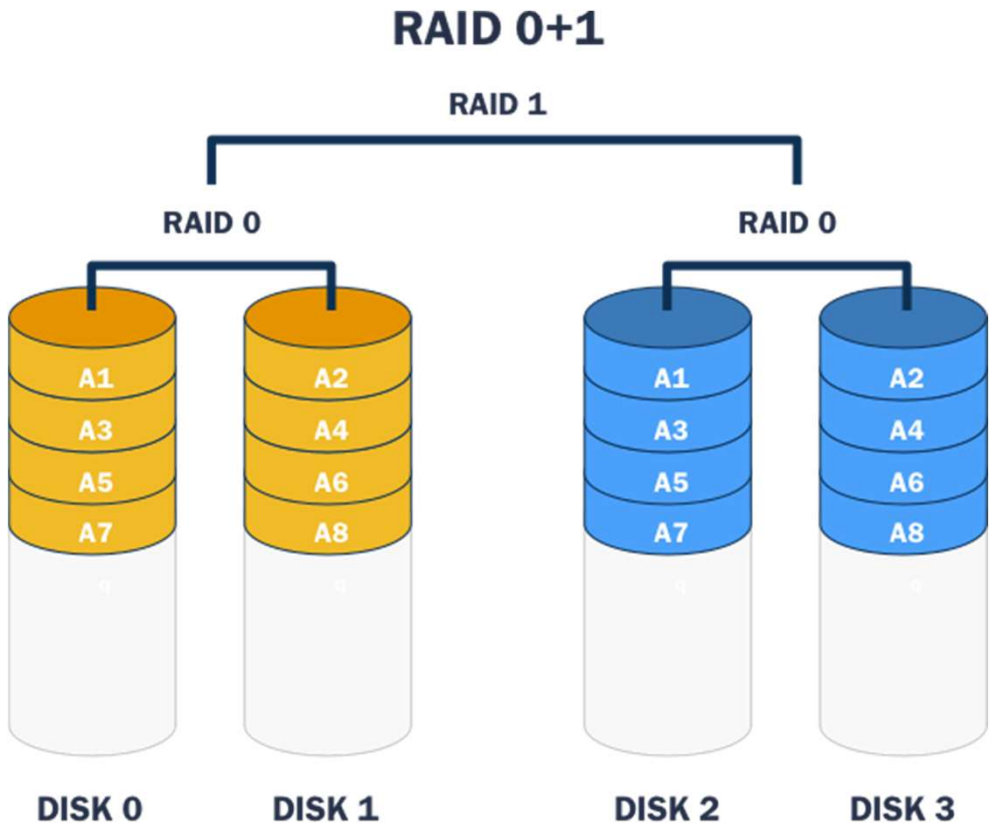
- Chaque système RAID **présente des avantages mais aussi des défauts.**
  ⇒ Il est possible de palier à ces défauts en **associant deux systèmes RAID différents** à plusieurs disques ou au contraire de profiter de leurs avantages.

- Le système le plus utilisé dans les combinaisons est **le RAID 0**. S'il est associé avec un système RAID de niveau 1, 4 ou 5, le système obtenu possède alors **un niveau de performance très élevé** ainsi qu'une **tolérance à la panne**

Principales combinaisons de RAID

La combinaison RAID 0 + RAID 1 (1):

- Un système **RAID 0+1** associe un **RAID 0** et un **RAID 1**. Le RAID 0 est implémenté en premier et le RAID 1 en second.
- Si on a une matrice de 10 disques, on divise les disques en deux groupes de 5, chaque groupe étant configuré en RAID 0. Les deux groupes sont alors reliés entre eux par un système RAID 1.



⇒ On applique donc la technique du mirroring sur un système de RAID 0.

La combinaison RAID 0 + RAID 1 (2):

- Si un disque tombe en panne, **alors on perd le RAID 1**, et on se retrouve avec **un simple système RAID 0**. De plus, si un disque de cette matrice tombe en panne, alors l'ensemble des données est perdu

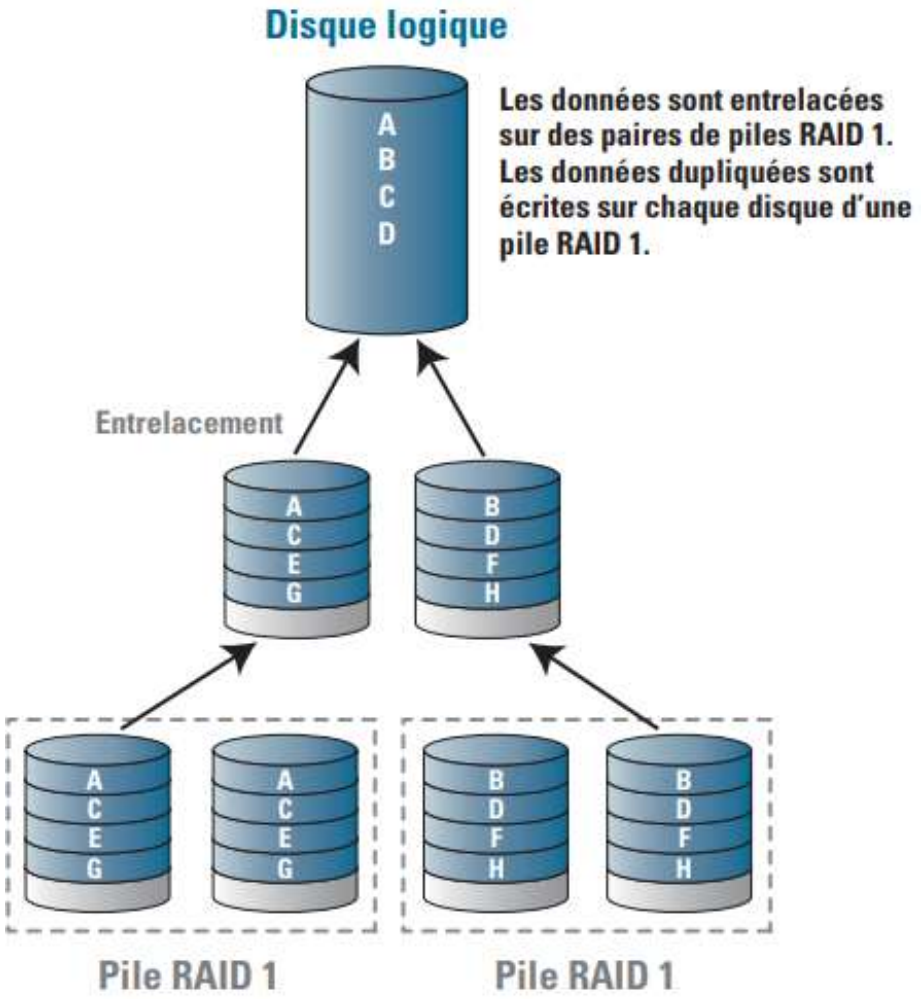
- Dans le cas d'une panne, temps de reconstitution des données est élevé. En effet, il faut rétablir les données des quatre disques. C'est pourquoi **le RAID 0+1** n'est que peu utilisé

Principales combinaisons de RAID

La combinaison RAID 1 + RAID 0 (1):

Ensemble de RAID 1 entrelacés

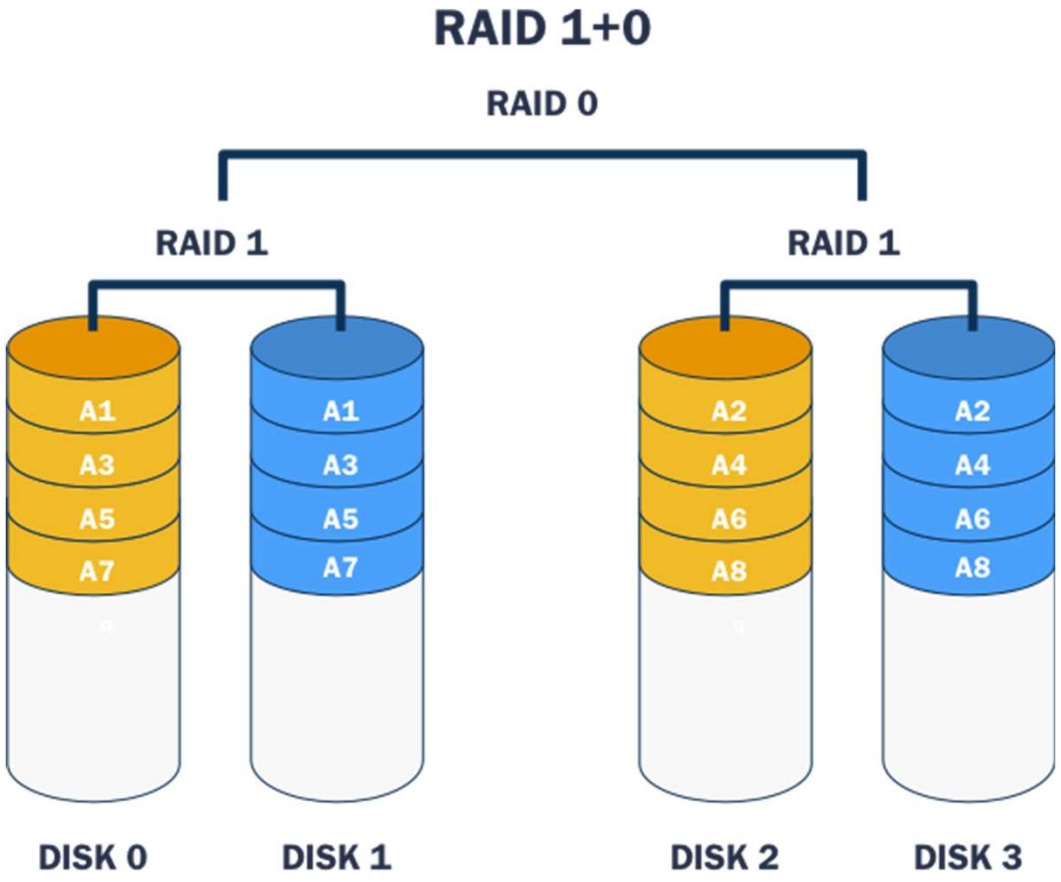
- Combine l'entrelacement **RAID 0** et l'écriture miroir **RAID 1**.
- Ce niveau offre l'optimisation des performances inhérente à **l'entrelacement des données** tout en assurant la redondance des données caractéristique de **l'écriture miroir**.
- Ce niveau RAID assure **la tolérance aux pannes** jusqu'à un disque par sous-pile peut tomber en panne sans entraîner de perte de données


Disque logique
A B C D
Les données sont entrelacées sur des paires de piles RAID 1. Les données dupliquées sont écrites sur chaque disque d'une pile RAID 1.
Entrelacement
A C E G
B D F H
A C E G
A C E G
B D F H
B D F H
Pile RAID 1
Pile RAID 1

Principales combinaisons de RAID

La combinaison RAID 1 + RAID 0 (2):

- Ce mode de RAID a toujours le même inconvénient que le RAID 1, c'est-à-dire **qu'il est coûteux en capacité de stockage à cause du mirroring.**
- Il nécessite **au minimum 4 disques** durs : le RAID 0 nécessite au moins 2 disques qui dans ce cas sont les 2 paires de disques du RAID 1.
- Etant donné **la baisse actuelle des prix des disques**, cette solution est de plus en plus envisagée.



Redundant Array of Independent Disks

Niveaux de RAID définis par l'Université de Berkeley :

- **RAID 0 :** Stripping (entrelacement de disques)
- **RAID 1** : Mirroring (miroir de disque)
- **RAID 1+0 (RAID 10)** : Stripping et Mirroring
- **RAID 4 :** Stripping sur plusieurs disques avec parité sur disque dédié
- **RAID 5 :** Stripping sur disques indépendants avec parité répartie

Types de RAIDs

- Il existe deux sortes de RAID :
  - ⇒ **Le RAID matériel** (les contrôleurs de disques gèrent eux-mêmes le RAID)
  - ⇒ **Le RAID logiciel** (il existe plusieurs solutions pour Linux, dont le pilote MD)

Le RAID logiciel (1)

- Dans ce cas, **le contrôle du RAID** est intégralement assuré par **une couche logicielle** du système d'exploitation. Cette couche s'intercale entre la couche d'abstraction matérielle (pilote) et la couche du système de fichiers.
- **Avantages:**
  - ⇒ peu cher ;
  - ⇒ très souple (administration logicielle) ;
  - ⇒ la grappe est compatible avec toutes les machines utilisant le même OS.
- **Inconvénients:**
  - ⇒ Réduction des **performances** du système d'exploitation
  - ⇒ Données relatives à la configuration du RAID sur le disque de démarrage : perte du système RAID en cas de défaillance de ce disque

Le RAID logiciel (2)

- **Implémentations:**
  - ⇒ Sous Windows XP et + seulement le **RAID 0** et **1** sont gérés et sous Windows Serveur le **RAID 5** est supporté ;
  - ⇒ Sous MAC seulement le **RAID 0** et **1** sont supportés ;
  - ⇒ Sous Linux (noyau 2.6) les **RAID 0, 1, 4, 5, 6 et 10** sont supportés ainsi que les combinaisons de ces modes (ex. **0+1**)

Le RAID matériel (1)

- Une carte ou **un composant** est dédié à la gestion des opérations, dotée d'un processeur spécifique, de mémoire, éventuellement d'une batterie de secours, et est capable de gérer tous les aspects du **système de stockage RAID**.

- **Avantages:**

  ⇒ **Détection des défauts**, remplacement à chaud des unités défectueuses, possibilité de reconstruire de manière transparente les disques défaillants ;

  ⇒ charge système **est allégée** ;

  ⇒ La vérification de cohérence, les diagnostics et les maintenances sont effectués en arrière-plan par le contrôleur **sans solliciter de ressources système.**

Le RAID matériel (2)

- **Inconvénients:**
  - ⇒ les contrôleurs RAID matériels utilisent chacun leur **propre système pour gérer les unités de stockage** et donc aucune donnée ne pourra être récupérée si le contrôleur RAID n'est pas exactement le même;
  - ⇒ couteux
  - ⇒ le contrôleur RAID est lui-même un composant matériel, qui **peut tomber en panne**

Choix du niveau de RAID

- Pour choisir **le niveau RAID** le mieux adapté à vos besoins, il faut prendre en compte un ensemble de facteurs. Il existe un rapport de compromis entre chacun de ces facteurs :
  - ⇒ **Coût du stockage** sur disques
  - ⇒ Niveau de **protection** ou de **disponibilité** des données requis (bas, moyen, élevé)
  - ⇒ **Niveau de performances** requis (bas, moyen, élevé)

Configuration

- **mdadm** (multiple devices admin) est un outil extrêmement utile pour créer, gérer et surveiller des périphériques RAID. Ces derniers peuvent êtres construits à partir des **disques entiers** ou des **partitions**.
- Les informations concernant les ensembles RAID en cours de fonctionnement sont accessibles avec le fichier **/proc/mdstat**
- Pour que les ensembles soient activés au démarrage il faut créer un fichier de configuration **/etc/mdadm/mdadm.conf**
- Pour que l'ensemble soit monté automatiquement au démarrage, il faut ajouter une ligne au fichier **/etc/fstab**

Conclusion

- Les systèmes RAID possèdent des atouts très important, c'est à dire **performance et fiabilité**.
- De plus, les coûts sont de plus en plus faible du fait de la baisse constante des prix des disques durs.
- Un système RAID **ne remplace en aucun cas à un système de sauvegarde**. Il ne protège en rien d'une erreur humaine.